

EMPro  
2015.01

# FEM Simulation

# Notice

---

## © Keysight Technologies, Inc. 2007-2015

1400 Fountaingrove Pkwy., Santa Rosa, CA 95403-1738, United States

No part of this documentation may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies, Inc. as governed by United States and international copyright laws.

## Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Java™ is a U.S. trademark of Sun Microsystems, Inc. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC.

## ARpack

BSD Software License

Pertains to ARPACK and P\_ARPACK

Copyright (c) 1996-2008 Rice University.

Developed by D.C. Sorensen, R.B. Lehoucq, C. Yang, and K. Maschhoff.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer listed in this license in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **CUDA Complex**

Copyright (c) 2008 Christian Buchner <[Christian.Buchner@gmail.com](mailto:Christian.Buchner@gmail.com)>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY Christian Buchner "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL Christian Buchner BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **DejaVu Fonts**

Fonts are (c) Bitstream (see below). DejaVu changes are in public domain. Glyphs imported from Arev fonts are (c) Tavmjung Bah (see below)

Bitstream Vera Fonts Copyright

-----

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the Gnome Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.

Arev Fonts Copyright

-----

Copyright (c) 2006 by Tavmjong Bah. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the modifications to the Bitstream Vera Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Tavmjong Bah" or the word "Arev".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Tavmjong Bah Arev" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL TAVMJONG BAH BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the name of Tavmjong Bah shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from Tavmjong Bah. For further information, contact: [tavmjong@free.fr](mailto:tavmjong@free.fr).

### **Google Breakpad**

Copyright (c) 2006, Google Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### **HOOMD**

Highly Optimized Object-Oriented Molecular Dynamics (HOOMD) Open Source Software License

Copyright (c) 2008 Ames Laboratory Iowa State University

All rights reserved.

Redistribution and use of HOOMD, in source and binary forms, with or without modification, are permitted, provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of the copyright holder nor the names HOOMD's contributors may be used to endorse or promote products derived from this software without specific prior written permission.

#### Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### Lass

Lass subject to the Common Public Attribution License Version 1.0 (the "License"); you may not use Lass except in compliance with the License. You may obtain a copy of the License at <http://lass.cocamware.com/cpal-license> or at [http://www.opensource.org/licenses/cpal\\_1.0](http://www.opensource.org/licenses/cpal_1.0).

The License is based on the Mozilla Public License Version 1.1 but Sections 14 and 15 have been added to cover use of software over a computer network and provide for limited attribution for the Original Developer. In addition, Exhibit A has been modified to be consistent with Exhibit B.

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is LASS - Library of Assembled Shared Sources. The Initial Developer of the Original Code is Bram de Greve and Tom De Muer. The Original Developer is the Initial Developer.

All portions of the code written by the Initial Developer are:

Copyright (C) 2004-2012 the Initial Developer.

All Rights Reserved.

Contributor(s):

Alternatively, the contents of this file may be used under the terms of the GNU General Public License Version 2 or later (the GPL), in which case the provisions of GPL are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the GPL and not to allow others to use your version of this file under the CPAL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the GPL License. If you do not delete the provisions above, a recipient may use your version of this file under either the CPAL or the GPL.

#### MPFR

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

#### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

#### 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.



b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

#### 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

#### **MPIR**

Copyright 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006  
Free Software Foundation, Inc.

Copyright 2009 William Hart

This file is part of the MPIR Library. The MPIR Library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

The MPIR Library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with the GNU MP Library; see the file COPYING.LIB. If not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

#### **PVM**

PVM version 3.4: Parallel Virtual Machine System

University of Tennessee, Knoxville TN.

Oak Ridge National Laboratory, Oak Ridge TN.

Emory University, Atlanta GA.

Authors: J. J. Dongarra, G. E. Fagg, G. A. Geist, J. A. Kohl, R. J. Manchek, P. Mucci, P. M. Papadopoulos, S. L. Scott, and V. S. Sunderam

(C) 2009 All Rights Reserved

NOTICE

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation.

Neither the Institutions (Emory University, Oak Ridge National Laboratory, and University of Tennessee) nor the Authors make any representations about the suitability of this software for any purpose. This software is provided "as is" without express or implied warranty.

PVM version 3 was funded in part by the U.S. Department of Energy, the National Science Foundation and the State of Tennessee.

## Python

### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <http://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <http://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <http://www.zope.com>). In 2001, the Python Software Foundation (PSF, see <http://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see <http://www.opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Release Derived Year Owner GPL- from compatible? (1)

0.9.0 thru 1.2 1991-1995 CWI yes

1.3 thru 1.5.2 1.2 1995-1999 CNRI yes

1.6 1.5.2 2000 CNRI no

2.0 1.6 2000 BeOpen.com no

1.6.1 1.6 2001 CNRI yes (2)

2.1 2.0+1.6.1 2001 PSF no

2.0.1 2.0+1.6.1 2001 PSF yes

2.1.1 2.1+2.0.1 2001 PSF yes

2.2 2.1.1 2001 PSF yes

2.1.2 2.1.1 2002 PSF yes

2.1.3 2.1.2 2002 PSF yes

2.2.1 2.2 2002 PSF yes

2.2.2 2.2.1 2002 PSF yes

2.2.3 2.2.2 2003 PSF yes

2.3 2.2.2 2002-2003 PSF yes

2.3.1 2.3 2002-2003 PSF yes

2.3.2 2.3.1 2002-2003 PSF yes

2.3.3 2.3.2 2002-2003 PSF yes

2.3.4 2.3.3 2004 PSF yes

2.3.5 2.3.4 2005 PSF yes

2.4 2.3 2004 PSF yes

2.4.1 2.4 2005 PSF yes

2.4.2 2.4.1 2005 PSF yes

2.4.3 2.4.2 2006 PSF yes

2.4.4 2.4.3 2006 PSF yes

2.5 2.4 2006 PSF yes

2.5.1 2.5 2007 PSF yes

2.5.2 2.5.1 2008 PSF yes

2.5.3 2.5.2 2008 PSF yes

2.6 2.5 2008 PSF yes

2.6.1 2.6 2008 PSF yes

2.6.2 2.6.1 2009 PSF yes

2.6.3 2.6.2 2009 PSF yes

2.6.4 2.6.3 2009 PSF yes

2.6.5 2.6.4 2010 PSF yes

2.7 2.6 2010 PSF yes

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

=====

### PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

-----

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

### BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

-----

## BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").
2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.
3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.
7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

## CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam,

The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

bzip2

-----

This copy of Python includes a copy of bzip2, which is licensed under the following terms:

-----

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2007 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, [jseward@bzip.org](mailto:jseward@bzip.org)

bzip2/libbzip2 version 1.0.5 of 10 December 2007

-----

## OpenSSL

-----

This copy of Python includes a copy of openssl, which is licensed under the following terms:

### LICENSE ISSUES

=====

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact [openssl-core@openssl.org](mailto:openssl-core@openssl.org).

### OpenSSL License

-----

/\* =====

\* Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\*

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in

\* the documentation and/or other materials provided with the

\* distribution.

\*

\* 3. All advertising materials mentioning features or use of this

\* software must display the following acknowledgment:

\* "This product includes software developed by the OpenSSL Project

\* for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"



\*

\* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to

\* endorse or promote products derived from this software without

\* prior written permission. For written permission, please contact

\* [openssl-core@openssl.org](mailto:openssl-core@openssl.org).

\*

\* 5. Products derived from this software may not be called "OpenSSL"

\* nor may "OpenSSL" appear in their names without prior written

\* permission of the OpenSSL Project.

\*

\* 6. Redistributions of any form whatsoever must retain the following

\* acknowledgment:

\* "This product includes software developed by the OpenSSL Project

\* for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

\*

\* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY

\* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

\* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR

\* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

\* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

\* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

\* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,

\* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

\* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED

\* OF THE POSSIBILITY OF SUCH DAMAGE.

\* =====

\*

\* This product includes cryptographic software written by Eric Young

\* ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)). This product includes software written by Tim

\* Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

\*

\*/

Original SSLeay License

-----

/\* Copyright (C) 1995-1998 Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com))

\* All rights reserved.

\*

\* This package is an SSL implementation written

\* by Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)).

\* The implementation was written so as to conform with Netscapes SSL.

\*

\* This library is free for commercial and non-commercial use as long as

\* the following conditions are aheared to. The following conditions

\* apply to all code found in this distribution, be it the RC4, RSA,

\* lhash, DES, etc., code; not just the SSL code. The SSL documentation

\* included with this distribution is covered by the same copyright terms

\* except that the holder is Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

\*

\* Copyright remains Eric Young's, and as such any Copyright notices in

\* the code are not to be removed.

\* If this package is used in a product, Eric Young should be given attribution

\* as the author of the parts of the library used.

\* This can be in the form of a textual message at program startup or

\* in documentation (online or textual) provided with the package.

\*

\* Redistribution and use in source and binary forms, with or without

- \* modification, are permitted provided that the following conditions
- \* are met:
- \* 1. Redistributions of source code must retain the copyright
- \* notice, this list of conditions and the following disclaimer.
- \* 2. Redistributions in binary form must reproduce the above copyright
- \* notice, this list of conditions and the following disclaimer in the
- \* documentation and/or other materials provided with the distribution.
- \* 3. All advertising materials mentioning features or use of this software
- \* must display the following acknowledgement:
- \* "This product includes cryptographic software written by
- \* Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com))"
- \* The word 'cryptographic' can be left out if the routines from the library
- \* being used are not cryptographic related :-).
- \* 4. If you include any Windows specific code (or a derivative thereof) from
- \* the apps directory (application code) you must include an acknowledgement:
- \* "This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com))"
- \*
- \* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND
- \* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- \* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- \* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
- \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- \* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- \* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- \* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
- \* SUCH DAMAGE.
- \*
- \* The licence and distribution terms for any publically available version or

\* derivative of this code cannot be changed. i.e. this code cannot simply be

\* copied and put under another distribution licence

\* [including the GNU Public Licence.]

\*/

Tcl

---

This copy of Python includes a copy of Tcl, which is licensed under the following terms:

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, ActiveState Corporation and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR

MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

Tk

--

This copy of Python includes a copy of Tk, which is licensed under the following terms:

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR

MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

Mersenne Twister

-----

The \_random module includes code based on a download from <http://www.math.keio.ac.jp/matsumoto/MT2002/emt19937ar.html>. The following are the verbatim comments from the original code:

A C-program for MT19937, with initialization improved 2002/1/26. Coded by Takuji Nishimura and Makoto Matsumoto.

Before using, initialize the state by using `init_genrand(seed)` or `init_by_array(init_key, key_length)`.

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Any feedback is very welcome.

<http://www.math.keio.ac.jp/matumoto/emt.html>

email: [matumoto@math.keio.ac.jp](mailto:matumoto@math.keio.ac.jp)

Sockets

-----

The socket module uses the functions, `getaddrinfo()`, and `getnameinfo()`, which are coded in separate source files from the WIDE Project, <http://www.wide.ad.jp/>.

Copyright (C) 1995, 1996, 1997, and 1998 WIDE Project.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Floating point exception control

-----  
The source for the fpectl module includes the following notice:

-----  
/ Copyright (c) 1996. \

| The Regents of the University of California. |

| All rights reserved. |

| |

| Permission to use, copy, modify, and distribute this software for |

| any purpose without fee is hereby granted, provided that this en- |

| tire notice is included in all copies of any software which is or |

| includes a copy or modification of this software and in all |

| copies of the supporting documentation for such software. |

| |

| This work was produced at the University of California, Lawrence |

| Livermore National Laboratory under contract no. W-7405-ENG-48 |

| between the U.S. Department of Energy and The Regents of the |

| University of California for the operation of UC LLNL. |

| |

| DISCLAIMER |

| |

| This software was prepared as an account of work sponsored by an |

| agency of the United States Government. Neither the United States |

| Government nor the University of California nor any of their em- |

| ployees, makes any warranty, express or implied, or assumes any |

| liability or responsibility for the accuracy, completeness, or |

| usefulness of any information, apparatus, product, or process |

| disclosed, or represents that its use would not infringe |

| privately-owned rights. Reference herein to any specific commer- |

| cial products, process, or service by trade name, trademark, |

| manufacturer, or otherwise, does not necessarily constitute or |  
| imply its endorsement, recommendation, or favoring by the United |  
| States Government or the University of California. The views and |  
| opinions of authors expressed herein do not necessarily state or |  
| reflect those of the United States Government or the University |  
| of California, and shall not be used for advertising or product |  
| endorsement purposes. /

-----  
MD5 message digest algorithm  
-----

The source code for the md5 module contains the following notice:

Copyright (C) 1999, 2002 Aladdin Enterprises. All rights reserved.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch

[ghost@aladdin.com](mailto:ghost@aladdin.com)

Independent implementation of MD5 (RFC 1321).

This code implements the MD5 Algorithm defined in RFC 1321, whose text is available at <http://www.ietf.org/rfc/rfc1321.txt> The code is derived from the text of the RFC, including the test suite (section A.5) but excluding the rest of Appendix A. It does not include any code or documentation that is identified in the RFC as being copyrighted.

The original and principal author of md5.h is L. Peter Deutsch <[ghost@aladdin.com](mailto:ghost@aladdin.com)>. Other authors are noted in the change history

that follows (in reverse chronological order):



2002-04-13 lpd Removed support for non-ANSI compilers; removed references to Ghostscript; clarified derivation from RFC 1321; now handles byte order either statically or dynamically. 1999-11-04 lpd Edited comments slightly for automatic TOC extraction. 1999-10-18 lpd Fixed typo in header comment (ansi2knr rather than md5); added conditionalization for C++ compilation from Martin Purschke <[purschke@bnl.gov](mailto:purschke@bnl.gov)>.

1999-05-03 lpd Original version.

Asynchronous socket services

-----

The asynchat and asyncore modules contain the following notice:

Copyright 1996 by Sam Rushing

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Sam Rushing not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

SAM RUSHING DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL SAM RUSHING BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Cookie management

-----

The Cookie module contains the following notice:

Copyright 2000 by Timothy O'Malley <[timo@alum.mit.edu](mailto:timo@alum.mit.edu)>

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Timothy O'Malley not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

Timothy O'Malley DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL Timothy O'Malley BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Profiling

-----

The profile and pstats modules contain the following notice:

Copyright 1994, by InfoSeek Corporation, all rights reserved.

Written by James Roskind

Permission to use, copy, modify, and distribute this Python software and its associated documentation for any purpose (subject to the restriction in the following sentence) without fee is hereby granted, provided that the above copyright notice appears in all copies, and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of InfoSeek not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. This permission is explicitly restricted to the copying and modification of the software to remain in Python, compiled Python, or other languages (such as C) wherein the modified or derived code is exclusively imported into a Python module.

INFOSEEK CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INFOSEEK CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Execution tracing

-----

The trace module contains the following notice:

portions copyright 2001, Autonomous Zones Industries, Inc., all rights... err... reserved and offered to the public under the terms of the Python 2.2 license.

Author: Zooko O'Whielacronx

<http://zooko.com/>

<mailto:zooko@zooko.com>

Copyright 2000, Mojam Media, Inc., all rights reserved.

Author: Skip Montanaro

Copyright 1999, Bioreason, Inc., all rights reserved.

Author: Andrew Dalke

Copyright 1995-1997, Automatrix, Inc., all rights reserved.

Author: Skip Montanaro

Copyright 1991-1995, Stichting Mathematisch Centrum, all rights reserved.

Permission to use, copy, modify, and distribute this Python software and its associated documentation for any purpose without fee is hereby granted, provided that the above copyright notice appears in all copies, and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of neither Automatrix, Bioreason or Mojam Media be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

UUencode and UUdecode functions

-----

The uu module contains the following notice:

Copyright 1994 by Lance Ellinghouse

Cathedral City, California Republic, United States of America.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Lance Ellinghouse not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

LANCE ELLINGHOUSE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL LANCE ELLINGHOUSE CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Modified by Jack Jansen, CWI, July 1995:

- Use binascii module to do the actual line-by-line conversion between ascii and binary. This results in a 1000-fold speedup. The C version is still 5 times faster, though.
- Arguments more compliant with Python standard XML Remote Procedure Calls

-----

The xmlrpclib module contains the following notice:

The XML-RPC client interface is

Copyright (c) 1999-2002 by Secret Labs AB

Copyright (c) 1999-2002 by Fredrik Lundh

By obtaining, using, and/or copying this software and/or its associated documentation, you agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its associated documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies, and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Secret Labs AB or the author not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

SECRET LABS AB AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANT- ABILITY AND FITNESS. IN NO EVENT SHALL SECRET LABS AB OR THE AUTHOR BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

test\_epoll

-----

The test\_epoll contains the following notice:

Copyright (c) 2001-2006 Twisted Matrix Laboratories.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Select kqueue

-----

The select and contains the following notice for the kqueue interface:

Copyright (c) 2000 Doug White, 2006 James Knight, 2007 Christian Heimes

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

strtod and dtoa

-----

The file Python/dtoa.c, which supplies C functions dtoa and strtod for conversion of C doubles to and from strings, is derived from the file of the same name by David M. Gay, currently available from <http://www.netlib.org/fp/>. The original file, as retrieved on March 16, 2009, contains the following copyright and licensing notice:

/\*\*\*\*\*

\*

\* The author of this software is David M. Gay.

\*

\* Copyright (c) 1991, 2000, 2001 by Lucent Technologies.

\*

\* Permission to use, copy, modify, and distribute this software for any

\* purpose without fee is hereby granted, provided that this entire notice

\* is included in all copies of any software which is or includes a copy

\* or modification of this software and in all copies of the supporting

\* documentation for such software.

\*

\* THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED

\* WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY

\* REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY

\* OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

\*

\*\*\*\*\*/

expat

-----

The pyexpat extension is built using an included copy of the expat sources unless the build is configured --with-system-expat:

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

libffi

-----

The \_ctypes extension is built using an included copy of the libffi sources unless the build is configured --with-system-libffi:

Copyright (c) 1996-2008 Red Hat, Inc and others.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

zlib

----

The zlib extension is built using an included copy of the zlib sources unless the zlib version found on the system is too old to be used for the build:

Copyright (C) 1995-2010 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly

Mark Adler

[jloup@gzip.org](mailto:jloup@gzip.org)

[madler@alumni.caltech.edu](mailto:madler@alumni.caltech.edu)

## Qt

Qt Copyright:

Qt Version 4.7.4, Copyright (c) 2010 by Nokia Corporation. All Rights Reserved.

Qt Notice:

The Qt code was modified. Used by permission.

Qt License:

Your use or distribution of Qt or any modified version of Qt implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included.

You may also contact Brian Buchanan at Keysight Technologies. at [brian\\_buchanan@keysight.com](mailto:brian_buchanan@keysight.com) for more information.

## RLog

Copyright (c) 2002-2004, Valient Gough

This library is free software; you can distribute it and/or modify it under the terms of the GNU Lesser General Public License (LGPL), as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the LGPL in the file COPYING for more details.

## **haslib++**

hashlib++ - a simple hash library for C++

Copyright (c) 2007,2008 Benjamin Grdelbach

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **pstdint**

A portable stdint.h

Copyright (c) 2005-2007 Paul Hsieh

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must not misrepresent the original source in the documentation and/or other materials provided with the distribution.

The names of the authors nor its contributors may be used to endorse or promote products derived from this software without specific prior written permission.



THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **pywin32**

Unless stated in the specific source file, this work is

Copyright (c) 1994-2008, Mark Hammond

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither name of Mark Hammond nor the name of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

win32com

-----

Unless stated in the specific source file, this work is Copyright (c) 1996-2008, Greg Stein and Mark Hammond.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither names of Greg Stein, Mark Hammond nor the name of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

adodbapi

-----

adodbapi - A python DB API 2.0 (PEP 249) interface to Microsoft ADO

Copyright (C) 2002 Henrik Ekelund, version 2.1 by Vernon Cole

\* <http://sourceforge.net/projects/pywin32>

\* <http://sourceforge.net/projects/adodbapi>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

django adaptations and refactoring by Adam Vandenberg isapi

-----

Copyright 2002-2003 by Blackdog Software Pty Ltd.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Blackdog Software not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

BLACKDOG SOFTWARE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL BLACKDOG SOFTWARE BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Scintilla and SciTE

-----

Copyright 1998-2003 by Neil Hodgson <[neilh@scintilla.org](mailto:neilh@scintilla.org)>

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

## **Errata**

The EMPro product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Keysight Technologies and is known as "Keysight EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

## **Warranty**

The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Keysight disclaims all warranties, either express or implied, with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Keysight shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Keysight and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

## **Technology Licenses**

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at <http://systemc.org/>. This software is redistributed by Keysight. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions of title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Keysight only.

### **Restricted Rights Legend**

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause.

Use, duplication or disclosure of Software is subject to Keysight Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

# Table of Contents

---

<b>FEM-based Simulations</b>	<b>48</b>
FEM Overview	48
The Finite Element Method	48
Representation of a Field Quantity	48
Field Quantities are Interpolated from Nodal Values	48
Basis Functions	49
Size of Mesh Versus Accuracy	49
Field Solutions	49
Implementation Overview	50
The Solution Process	50
Adaptive Solution	50
Non-adaptive Discrete Frequency Sweep	50
Non-adaptive Fast Frequency Sweep	51
The Mesher	51
2D Mesh Refinement	51
Modes	51
Modes, Reflections, and Propagation	51
Modes and Frequency	52
Modes and Multiple Ports on a Face	52
The 3D Solver	52
Boundary Conditions	53
Port Boundaries	53
Absorbing (Radiation)	53
Computing Radiated Fields	54
Implementing Green's Function When Computing Radiated Fields	55
Displaying Field Solutions	55
Ports-only Solutions and Impedance Computations	55
Application Areas	57
Visualization and Display of Results	57
FEM Simulator Process Overview	57
New FEM Low Frequency Algorithm	58
Eigenmode Overview	59
Tetrahedron sampling	59
Materials and Boundary Conditions	61

Eigenmode Matrix System .....	61
Mesh Accuracy and Refinement .....	61
Result Display .....	62
FEM 2D Port Solver Overview .....	62
Excitation Fields .....	62
Wave Equation .....	63
Modes .....	63
Mode Conversion .....	63
Modes, Reflections, and Propagation .....	63
Modes and Frequency .....	64
Displaying Fields by Mode .....	64
<b>Specifying FEM Simulation Setup .....</b>	<b>65</b>
Selecting the FEM Simulator .....	65
Reusing the Simulation Results .....	65
Defining FEM Simulation Options .....	65
Completing the Process of Specifying Setup .....	66
Specifying Frequency Plans in FEM .....	66
Sweep Type .....	66
Start Frequency .....	67
Sample Points Limit .....	67
Stop Frequency .....	67
Defining a Frequency Plan .....	67
Selecting a Field Storage Option .....	67
Specifying Mesh and Refinement Properties in FEM .....	68
Stop Criterium Tab Settings .....	69
Refinement Tab Settings .....	70
Initial Mesh Tab Settings .....	71
Specifying Advanced Tab Settings .....	72
Vertex Mesh .....	73
Edge Mesh .....	74

Specifying Mesh Refinement Properties in FEM .....	75
Stop Criterium Tab Settings .....	75
Refinement Tab Settings .....	76
Initial Mesh Tab Settings .....	77
Specifying Advanced Tab Settings .....	78
Vertex Mesh .....	79
Edge Mesh .....	80
Selecting a Solver in FEM .....	81
Order of Basis Functions Setting .....	81
<b>Specifying FEM 2D Port Simulation Setup .....</b>	<b>83</b>
Adding Waveguide Ports .....	83
Selecting FEM 2D Port Simulation .....	84
Selecting Sources .....	85
Specifying the Frequency Plans .....	86
Defining a Frequency Plan .....	86
Specifying Convergence Properties .....	87
Adding Notes .....	87
Completing the Process of Specifying Setup .....	87
Running FEM 2D Port Simulation .....	87
Running FEM 2D Port Simulation from the EMPro Waveguide Ports Editor Dialog Box .....	88
<b>Specifying Eigenmode Simulation Setup .....</b>	<b>89</b>
Eigenmode Overview .....	89
Selecting the Eigenmode Solver .....	89
Specifying Frequency Plans .....	90
Setting Frequencies .....	90
Specifying Mesh Refinement .....	92
Stop Criterium Tab Settings .....	92
Initial mesh Tab Settings .....	92
Advanced Tab Settings .....	93

Guidelines . . . . .	94
Setting Mesh Refinement Options . . . . .	94
Adding Notes . . . . .	96
Completing the Process of Specifying Setup . . . . .	96
Running Eigenmode Simulation . . . . .	96
<b>Creating a Mesh . . . . .</b>	<b>97</b>
Initial Mesh Settings . . . . .	97
Target Mesh Options . . . . .	97
Conductor Mesh Options . . . . .	98
Further Customizing Mesh Lengths . . . . .	99
Meshing Properties Editor . . . . .	100
Group Application of Mesh Settings . . . . .	100
Edge Meshing . . . . .	100
Vertex Meshing . . . . .	101
Surface Meshing . . . . .	102
Combined Edge and Surface Meshing . . . . .	103
General Settings and GUI . . . . .	104
Using Initial Mesh Options . . . . .	104
Initial Target Mesh Size . . . . .	104
Initial Minimum Mesh Size . . . . .	105
FEM Meshing Options . . . . .	106
Meshing Properties Editor . . . . .	107
Edge Meshing . . . . .	107
Vertex Meshing . . . . .	108
Surface Meshing . . . . .	109
Combined Edge and Surface Meshing . . . . .	110
General Settings and GUI . . . . .	111
Edge and Vertex Meshing Guidelines . . . . .	111
Adaptive vs Initial Meshing . . . . .	111



Guidelines for Specifying Mesh Settings .....	112
Adaptive vs Initial Meshing .....	112
Target mesh size .....	112
Minimal Mesh Size .....	113
Merge Objects with Same Material .....	113
Using Conductor Meshing .....	113
Choosing the Edge and Vertex mesh values .....	114
Coaxial Structures Fed by a Waveguide .....	114
Performing FEM Broadband Refinement .....	115
Importance of Refinement Frequencies .....	116
Performing Manual Broadband Refinement .....	116
Performing Automatic Broadband Refinement .....	116
Multiple Fixed Refinement Frequencies .....	117
Refinement Frequencies chosen Automatically After Initial Pass .....	117
Refinement Frequencies Chosen Automatically After Each Pass .....	118
Selecting a Refinement Strategy .....	118
Viewing FEM Mesh .....	119
Show me How Do I View Mesh .....	120
Controlling the Mesh View .....	120
Viewing a Shaded Mesh .....	120
Viewing a Background Mesh .....	121
Viewing a Mesh for Boundary Faces .....	121
Viewing a Mesh for Port Faces .....	122
Reusing Mesh and Frequency Points .....	122
Using the Existing Simulation Setup Settings .....	122
Troubleshooting Mesh Failures .....	123
Best practices .....	123
Known Issue .....	124
Using English units may create small gaps or slivers in some designs .....	124
Solution .....	124

<b>Using Ports in FEM Simulation</b>	<b>125</b>
Waveguide Ports (2D-plane source)	125
Internal ports	125
Internal sheet ports	125
Circuit Component Port	126
Component Feeds	126
Waveguide Port	128
Internal Sheet Ports with Reduced Parasitics	128
 <b>Creating Waveguide Ports</b>	 <b>129</b>
Contents	129
Setting up Waveguide Ports	129
Creating a Waveguide Port	130
Guidelines for Waveguide Port Size	131
Specifying Port Dimensions	132
Using the Auto-extend Option	133
Specifying Port Properties	133
Specifying Waveguide Port Definition	134
Nodal vs Modal Ports	135
Number of Modes	136
Defining Reference Offsets	136
Why Use Reference Offsets?	136
Creating an Impedance Line	138
About Impedance Lines	138
Creating Impedance Lines	138
 <b>Calculating S-Parameters in FEM Simulation</b>	 <b>141</b>
Frequency Points	141
Z- and Y-Matrices	142
Characteristic Impedances	142
PI Impedance	142

PV Impedance .....	143
VI Impedance .....	143
Choice of Impedance .....	143
De-embedding .....	143
Equations .....	144
Derivation of Wave Equation .....	144
Maxwell Equations .....	145
Phasor Notation .....	145
Electric and Magnetic Fields Can Be Represented as Phasors .....	147
Assumptions .....	147
Conductivity .....	147
Dielectric Loss Tangent .....	147
Magnetic Loss Tangent .....	147
Definition of Freespace Phase Constant .....	148
<b>Using FEM RLC Loads .....</b>	<b>149</b>
Adding a Passive Load Component .....	149
Adding a Passive Load Definition .....	150
Adding a Feed Definition .....	151
<b>Hybrid FEM-MoM Boundary Conditions .....</b>	<b>153</b>
<b>Examples How to Simulate using FEM .....</b>	<b>155</b>
Create and Simulate a Microstrip Line .....	155
Create an EMPro Project .....	155
Creating the Microstrip line Geometry .....	155
Creating a Substrate .....	155
Creating a Microstrip Line .....	157
Defining Mesh Priority .....	159
Creating and Assigning Materials .....	159
Assigning Materials .....	160
Defining the Outer Boundary .....	161
Adding a Waveguide Port .....	161
Adding a Port .....	161
Define Voltage .....	162
Define Impedance line .....	162
Setting up FEM Simulations .....	163

Viewing Results .....	163
Viewing Planar Results .....	163
Advanced Visualization .....	164
Simulating a Microstrip Line with Sheet Port .....	165
Opening the Microstrip 50 Ohm Project .....	165
Adding Sheet to Port .....	166
Setting up an FEM Simulation .....	167
Comparing Results .....	168
Simulating a Microstrip Line with Symmetric Plane .....	169
Opening the Microstrip 50 Ohm Line Project .....	169
Modifying Microstrip Line Geometry .....	169
Applying Symmetry Boundary Conditions .....	172
Setting up an FEM Simulation .....	173
Visualizing Symmetric Plane .....	176
Performing Multimode Analysis on Rectangular Waveguide .....	177
Project Setup .....	178
Simulation Setup .....	178
Results .....	178
Step 1: Setting up Waveguide Geometry Model .....	178
Setting up Waveguide Ports .....	179
Setting up Simulation .....	182
Viewing Results .....	183
Viewing S Parameter .....	183
Viewing Propagation Constant .....	184
Viewing Field .....	184
Performing an Eigenmode Simulation for a Rectangular Cavity .....	187
Creating a New Project .....	187
Creating a New Geometry .....	188
Creating and Assigning Materials .....	188
Defining the Outer Boundary .....	188
Setting up an Eigenmode Simulation .....	190
Running the Eigenmode Simulation .....	190

Viewing Eigenmode Simulation Results ..... 190

    Adding Lossy Dielectric ..... 190

    Adding Lossy Metal ..... 190

The Finite Element Method (FEM) Simulator provides a complete solution for electromagnetic simulation of arbitrarily-shaped and passive three-dimensional structures. It provides a complete 3D EM simulation for designers working with RF circuits, MMIC, PC boards, modules, and Signal Integrity applications. Developed with the designer of high-frequency/high-speed circuits, FEM Simulator offers a powerful finite-element EM simulator that solves a wide array of applications with impressive accuracy and speed.

The FEM simulation technology provides the following features:

- Conductors, isotropic dielectrics, isotropic linear magnetic material modeling that allow a wide range of application coverage.
- An unlimited number of ports, which enables simulating multi I/O design applications such as packages.
- Electric and magnetic fields modeling, enabling visualization of EM fields in a design.
- Absorbing boundary condition (free space), enabling antenna modeling.
- Full-wave, EM-accuracy for first-pass design success.
- Antenna parameters (such as gain, directivity, and polarization), to enable better insight into antenna design.
- FEM Simulator and ADS integration provides an integrated approach to EM and Circuit designs.

This section provides information about the following topics:

- **FEM-based Simulations**
  - **FEM Overview**
  - **Eigenmode Overview**
  - **FEM 2D Port Solver Overview**
- **Specifying FEM Simulation Setup**
  - **Specifying Frequency Plans in FEM**
  - **Specifying Mesh and Refinement Properties in FEM**
  - **Selecting a Solver in FEM**
  - **Remote and Distributed Simulation**
- **Specifying FEM 2D Port Simulation Setup**
- **Specifying Eigenmode Simulation Setup**
- **Creating a Mesh**
  - **Initial Mesh Settings**
  - **Guidelines for Specifying Mesh Settings**
  - **Performing FEM Broadband Refinement**
  - **Viewing FEM Mesh**
  - **Reusing Mesh and Frequency Points**
  - **Troubleshooting Mesh Failures**
- **Using Ports in FEM Simulation**

- Creating Waveguide Ports
  - Setting up Waveguide Ports
  - Specifying Port Dimensions
  - Specifying Port Properties
  - Creating an Impedance Line
- Calculating S-Parameters in FEM Simulation
- Using FEM RLC Loads
- Viewing FEM Simulation Results
- Viewing Eigenmode Simulation Results
- Advanced Visualization of FEM Results
  - Specifying Visualization Properties
  - Controlling Visualization Excitations
  - Setting up Plots
  - Viewing the Far-Field Pattern of a Multi-Port Antenna
  - Exporting Field Data
- Hybrid FEM-MoM Boundary Conditions
- Examples How to Simulate using FEM

# FEM-based Simulations

- [FEM Overview](#)
- [Eigenmode Overview](#)
- [FEM 2D Port Solver Overview](#)

## FEM Overview

The FEM simulation technique used to calculate the full three-dimensional electromagnetic field inside a structure is based on the finite element method. This section provides an overview of the FEM simulation process, its implementation in EMPro, and a description of how S-parameters are computed from the simulated electric and magnetic fields.

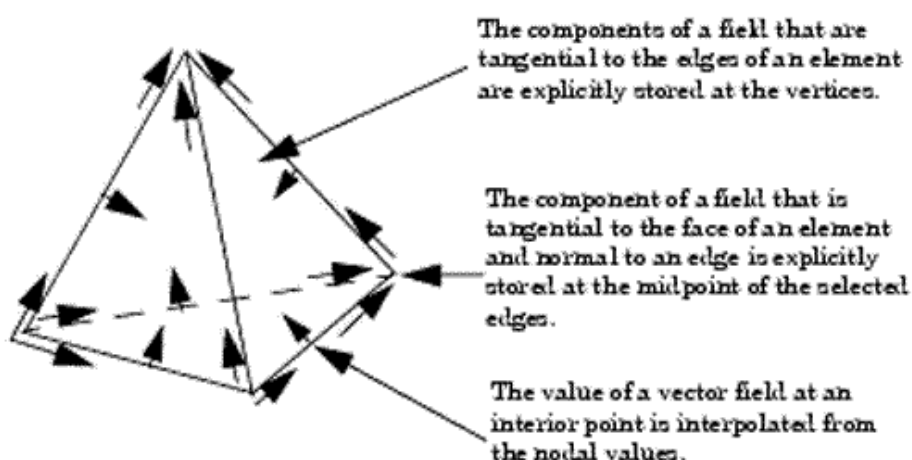
### The Finite Element Method

The finite element method divides the full problem space into many smaller regions (elements) and represents the field in each element by a local function. The FEM implementation in EMPro uses tetrahedral elements.

### Representation of a Field Quantity

The value of a vector field quantity (such as the H-field or the E-field) at points inside each tetrahedron is interpolated from the vertices of the tetrahedron. At each vertex, FEM Simulator stores the components of the field that are tangential to the three edges of the tetrahedron. In addition, the component of the vector field at the midpoint of selected edges that is tangential to a face and normal to the edge can also be stored. The field inside each tetrahedron is interpolated from these nodal values.

### Field Quantities are Interpolated from Nodal Values





The components of a field that are tangential to the edges of an element are explicitly stored at the vertices.

The component of a field that is tangential to the face of an element and normal to an edge is explicitly stored at the midpoint of the selected edges.

The value of a vector field at an interior point is interpolated from the nodal values.

By representing field quantities in this way, Maxwell's equations can be transformed into matrix equations that are solved using traditional numerical methods.

## Basis Functions

A first-order tangential element basis function interpolates field values from vector functions associated with edges.

First-order tangential elements have 6 unknowns per tetrahedron. Second-order tangential basis functions interpolate field values from vector functions associated with edges and faces. Second-order tangential elements have 20 unknowns per tetrahedron.

Order of Basis Function	Polynomial Model of E-fields	Polynomial Model of H-fields	Number of Unknowns per Tetrahedron
1	Linear	Constant	6
2	Quadratic	Linear	20

## Size of Mesh Versus Accuracy

There is a trade-off between the size of the mesh, the desired level of accuracy, and the amount of available computing resources.

On one hand, the accuracy of the solution depends on how small each of the individual elements (tetrahedra) are.

Solutions based on meshes that use a large number of elements are more accurate than solutions based on coarse meshes using relatively few elements. To generate a precise description of a field quantity, each tetrahedron must occupy a region that is small enough for the field to be adequately interpolated from the nodal values.

On the other hand, generating a field solution for meshes with a large number of elements requires a significant amount of computing power and memory. Therefore, it is desirable to use a mesh that is fine enough to obtain an accurate field solution but not so fine that it overwhelms the available computer memory and processing power.

To produce the optimal mesh, FEM Simulator uses an iterative process in which the mesh is automatically refined in critical regions. First, it generates a solution based on a coarse initial mesh. Then, it refines the mesh based on suitable error criteria and generates a new solution. When selected S-parameters converge to within a desired limit, the iteration process ends.

## Field Solutions

During the iterative solution process, the S-parameters typically stabilize before the full field solution. Therefore, when you are interested in analyzing the field solution associated with a structure, it may be desirable to use convergence criteria that is tighter than usual.

In addition, for any given number of adaptive iterations, the magnetic field (H-field) is less accurate than the solution for the electric field (E-field) because the H-field is computed from the E-field using the following relationship:

$$\mathbf{H} = \frac{\nabla \times \mathbf{E}}{-j\omega\mu}$$

This lowers the polynomial interpolation function than those used for the electric field.

## Implementation Overview

To calculate the S-matrix associated with a structure, the following steps are performed:

1. The structure is divided into a finite element mesh.
2. The waves on each port of the structure that are supported by a transmission line having the same cross section as the port are computed.
3. The full electromagnetic field pattern inside the structure is computed, assuming that each of the ports is excited by one of the waves.
4. The generalized S-matrix is computed from the amount of reflection and transmission that occurs.

The final result is an S-matrix that allows the magnitude of transmitted and reflected signals to be computed directly from a given set of input signals, reducing the full three-dimensional electromagnetic behavior of a structure to a set of high frequency circuit values.

## The Solution Process

There are three variations to the solution process:

- Adaptive solution
- Non-adaptive discrete frequency sweep
- Non-adaptive fast frequency sweep

### Adaptive Solution

An adaptive solution is one in which a finite element mesh is created and automatically refined to increase the accuracy of succeeding adaptive solutions. The adaptive solution is performed at a single frequency. (Often, this is the first step in generating a non-adaptive frequency sweep or a fast frequency sweep).

### Non-adaptive Discrete Frequency Sweep

To perform this type of solution, an existing mesh is used to generate a solution over a range of frequencies. You specify the starting and ending frequency, and the interval at which new solutions are generated. The same mesh is used for each solution, regardless of the frequency.

## Non-adaptive Fast Frequency Sweep

This type of solution is similar to a discrete frequency sweep, except that a single field solution is performed at a specified center frequency. From this initial solution, the system employs asymptotic waveform evaluation (AWE) to extrapolate an entire bandwidth of solution information. While solutions can be computed and viewed at any frequency, the solution at the center frequency is the most accurate.

## The Mesher

A mesh is the basis from which a simulation begins. Initially, the structure's geometry is divided into a number of relatively coarse tetrahedra, with each tetrahedron having four triangular faces. The mesher uses the vertices of objects as the initial set of tetrahedra vertices. Other points are added to serve as the vertices of tetrahedra only as needed to create a robust mesh. Adding points is referred to as seeding the mesh.

After the initial field solution has been created, if adaptive refinement is enabled, the mesh is refined further.

## 2D Mesh Refinement

For 2D objects or ports, the mesher treats its computation of the excitation field pattern as a two-dimensional finite element problem. The mesh associated with each port is simply the 2D mesh of triangles corresponding to the face of tetrahedra that lie on the port surface.

The mesher performs an iterative refinement of this 2D mesh as follows:

1. Using the triangular mesh formed by the tetrahedra faces of the initial mesh, solutions for the electric field,  $\mathbf{E}$ , are calculated.
2. The 2D solution is verified for accuracy.
3. If the computed error falls within a pre-specified tolerance, the solution is accepted. Otherwise, the 2D mesh on the port face is refined and another iteration is performed.

Any mesh points that have been added to the face of a port are incorporated into the full 3D mesh.

## Modes

For a waveguide or transmission line with a given cross section, there is a series of basic field patterns (modes) that satisfy Maxwell's equations at a specific frequency. Any linear combination of these modes can exist in the waveguide.

## Modes, Reflections, and Propagation

It is also possible for a 3D field solution generated by an excitation signal of one specific mode to contain reflections of higher-order modes which arise due to discontinuities in a high frequency structure. If these higher-order modes are reflected back to the excitation port or transmitted onto another port, the S-parameters associated with these modes should be calculated.

If the higher-order mode decays before reaching any port—either because of attenuation due to losses or because it is a non-propagating evanescent mode—there is no need to obtain the S-parameters for that mode. Therefore, one way to avoid the need for computing the S-parameters for a higher-order mode is to include a length of waveguide in the geometric model that is long enough for the higher-order mode to decay.

For example, if the mode 2 wave associated with a certain port decays to near zero in 0.5 mm, then the "constant cross section" portion of the geometric model leading up to the port should be at least 0.5 mm long. Otherwise, for accurate S-parameters, the mode 2 S-parameters must be included in the S-matrix.

The length of the constant cross section segment to be included in the model depends on the value of the mode's attenuation constant, .

## Modes and Frequency

The field patterns associated with each mode generally vary with frequency. However, the propagation constants and impedance always vary with frequency. Therefore, when a frequency sweep has been requested, a solution is calculated for each frequency point of interest.

When performing frequency sweeps, be aware that as the frequency increases, the likelihood of higher-order modes propagating also increases.

## Modes and Multiple Ports on a Face

Visualize a port face on a microstrip that contains two conducting strips side by side as two separate ports. If the two ports are defined as being separate, they are treated as two ports are connected to uncoupled transmission structures. It is as if a conductive wall separates the excitation waves.

However, in actuality, there will be electromagnetic coupling between the two strips. The accurate way to model this coupling is to analyze the two ports as a single port with multiple modes.

## The 3D Solver

To calculate the full 3D field solution, the following wave equation is solved:

$$\nabla \times \left( \frac{1}{\mu_r} \nabla \times \mathbf{E}(x,y,z) \right) - k_0^2 \epsilon_r \mathbf{E}(x,y,z) = 0$$

where:

- $\mathbf{E}(x,y,z)$  is a complex vector representing an oscillating electric field.
- $\mu_r(x,y)$  is the complex relative permeability.
- $k_0$  is the free space phase constant,  $\omega \sqrt{\mu_0 \epsilon_0}$ ,
- $\omega$  is the angular frequency,  $2\pi f$ .
- $\epsilon_r(x,y)$  is the complex relative permittivity.

This is the same equation that the 2D solver solves for in calculating the 2D field pattern at each port. The difference is that the 3D solver does not assume that the electric field is a traveling wave propagating in a single direction. It assumes that the vector  $\mathbf{E}$  is a function of  $x$ ,  $y$ , and  $z$ . The physical electric field,  $\mathbf{E}(x,y,z,t)$ , is the real part of the product of the phasor,  $\mathbf{E}(x,y,z)$ , and  $j\omega t$ :

$$\mathbf{E}(x,y,z,t) = \text{Re}[\mathbf{E}(x,y,z)e^{j\omega t}]$$

## Boundary Conditions

FEM Simulator imposes boundary conditions at all surfaces exposed to the edge of the meshed problem region. This includes all outer surfaces and all surfaces exposed to voids and surface discontinuities within the structure.

The following types of boundary conditions are recognized by the 3D solver:

- Port
- Absorbing (Radiation)
- PEC (Perfect Electric Conductor)
- PMC (Perfect Magnetic Conductor)
- ESymmetry (Electric Symmetry condition - odd symmetry of tangential E-fields)
- MSymmetry (Magnetic Symmetry condition - even symmetry of tangential E-fields)

## Port Boundaries

The 2D field solutions generated by the 2D solver for each port serve as boundary conditions at those ports. The final field solution that is computed for the structure must match the 2D field pattern at each port.

FEM Simulator solves several problems in parallel. Consider the case of analyzing modes 1 and 2 in a two-port device. To compute how much of a mode 1 excitation at port 1 is transmitted as a mode 2 wave at port 2, the 3D mesher uses the following as boundary conditions:

- A "mode 1" field pattern at port 1.
- A "mode 2" field pattern at port 2.

To compute the full set of S-parameters, solutions involving other boundary conditions must also be solved. Because the S-matrix is symmetric for reciprocal structures (that is,  $S_{12}$  is the same as  $S_{21}$ ), only half of the S-parameters need to be explicitly computed.

## Absorbing (Radiation)

Radiation boundaries model surfaces that represent open space. Energy is allowed to radiate from these boundaries instead of being contained within them. At these surfaces, the second order radiation boundary condition is employed:

$$(\nabla \times \mathbf{E})_{\text{tan}} = jk_0 \mathbf{E}_{\text{tan}} - \frac{j}{k_0} \nabla \times \hat{\mathbf{n}} (\nabla \times \mathbf{E})_{\mathbf{n}} + \frac{j}{k_0} \nabla_{\text{tan}} (\nabla_{\text{tan}} \cdot \mathbf{E}_{\text{tan}})$$

where:

- $E_{\text{tan}}$  is the component of the E-field that is tangential to the surface.
- $\hat{\mathbf{n}}$  is the unit vector normal to the radiation surface.
- $k_0$  is the free space phase constant,  $\omega \sqrt{\mu_0 \epsilon_0}$ .
- $j$  is equal to  $\sqrt{-1}$ .

To ensure accurate results, radiation boundaries should be applied at least one quarter of a wavelength away from the source of the signal. However, they do not have to be spherical. The only restriction regarding their shape is that they be convex with regard to the radiation source.

## Computing Radiated Fields

Electromagnetic Design System maps the E-field computed by the 3D solver on the radiation surfaces to plane registers and then calculates the radiated E-field using the following equation:

$$\mathbf{E}(x,y,z) = \int_s ((j\omega\mu_0 \mathbf{H}_{tan})G + (\mathbf{E}_{tan} \times \nabla G) + (\mathbf{E}_{normal} \nabla G)) ds$$

where:

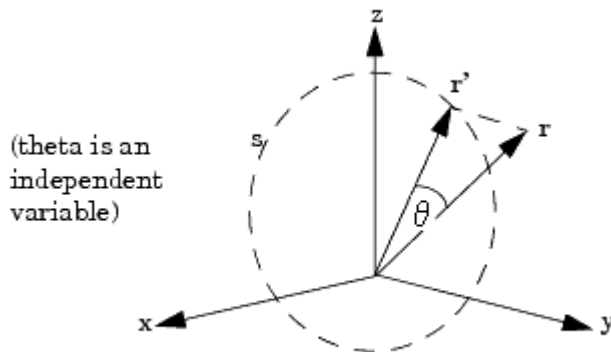
- $s$  represents the radiation surfaces.
- $j$  is the imaginary unit,  $\sqrt{-1}$ .
- $\omega$  is the angular frequency,  $2\pi f$ .
- $\mu_0$  is the relative permeability of the free space.
- $H_{tan}$  is the component of the magnetic field that is tangential to the surface.
- $H_{normal}$  is the component of the magnetic field that is normal to the surface.
- $E_{tan}$  is the component of the electric field that is tangential to the surface.
- $G$  is the free space Green's Function, given by:

$$G = \frac{e^{-jk_0|\vec{r}-\vec{r}'|}}{|\vec{r}-\vec{r}'|}$$

where:

- $k_0$  is the free space wave number,  $\omega\sqrt{\mu_0\epsilon_0}$ .
- $r$  and  $\vec{r}'$  represent, respectively.

## Implementing Green's Function When Computing Radiated Fields



## Displaying Field Solutions

The 3D solver is also used to manipulate field quantities for display. The system enables you to display or manipulate the field associated with any excitation wave at any port—for example, the field inside the structure due to a discrete mode 2 excitation wave at port 3. Waves excited on different modes can also be superimposed, even if they have different magnitudes and phases—for example, the waves excited on mode 1 at port 1 and mode 2 at port 2. In addition, far-field radiation in structures with radiation boundaries can be displayed.

The available fields depend on the type of solution that was performed:

- For adaptive solutions, the fields associated with the solution frequency are available.
- For frequency sweeps, the fields at each solved frequency point are available.
- For fast frequency sweeps, the fields associated with the center frequency point are initially available.

## Ports-only Solutions and Impedance Computations

This section addresses how impedances are computed for multi-conductor transmission line ports. Some examples of such structures are:

- Two coupled microstrip lines
- Coplanar waveguide modeled with 3 separate strips
- Shielded twin-wire leads

For structures with one or two conductors, you will need to define a single line segment, called an "impedance line", for each mode. Some examples of such structures include:

- microstrip transmission line (two-conductor structure)
- grounded CPW (two-conductor structure) where the CPW ground fins are also attached to the *OUTER* ground

For these structures, the port solver will compute the voltage  $V$  along the impedance line which is used to calculate  $Z_{pv} = V^2 / (2 * \text{Power})$ . The power is always normalized to 1 Watt.

If one models N-conductor structures where  $N > 2$ , then FEM Simulator uses a different algorithm for computing  $Z_{pv}$  and  $Z_{pi}$ . The user must define an impedance line for each interior conductor. The impedance lines should go from the center of each interior conductor to the outer conductor.

The port solver computes a voltage along the first N line segments for each of the first N modes when the port solver

detects that there are N+1 conductors. This becomes a "voltage vector"  $\bar{V}$  (of length N) for each of the N quasi-TEM modes. Then, when computing  $Z_{pv}$ , the square of the scalar voltage is now replaced by the dot product of the voltage vectors, for example:

$$Z_{pv} = \bar{V} \bullet \bar{V} / (2 \cdot Power)$$

For a more detailed explanation as to why this is done, refer to reference "1" at the end of this section.

For  $Z_{pi}$ , the current is generally computed by adding the currents flowing into and out of the port and taking the average of the two. (If the simulator computed currents to perfect accuracy, the inward and outward currents would be identical.)

For all mode numbers  $> N$ , where  $N$  = number of conductors, the currents are calculated in this way.

For the first (N-1) quasi-TEM modes, the currents are computed on the N-1 interior conductors producing a current eigenvector. Then the impedance:

$$Z_{pi} = (2 \cdot Power) / (\bar{I} \bullet \bar{I})$$

The result is that the FEM Simulator impedance computations for such structures as coupled microstrip lines match the published equations for even- and odd-mode impedances.

As an example, take a CPW modeled as three interior strips surrounded by an enclosure. The ground strips do not touch the enclosure. Such a model is in the examples directory of FEM Simulator and is called *cpwtaper*. The port solver shows us that the desired CPW mode is not the dominant mode, but is actually mode 3. To identify the modes, one can use the arrow plots in the Port Calibration menu or the Arrow display of the E-field in the post processor.

Each port consists of a 4 conductor system, the outer (ground) conductor, the inner strip, and the two "ground" strips. This results in 3 quasi-TEM modes. Mode 1 has E-field lines predominately in the substrate, all pointing in the same direction. This is the common mode (+V, +V, +V). Mode 2 also has E-field lines predominately in the substrate, but in opposite directions under the two "ground" strips. This is the slot mode (-V, 0, +V). Mode 3 has nearly zero E-fields everywhere because the fields are predominately between the inner strip and the "ground" strips. This is the CPW mode (0, +V, 0).

For further help in identifying modes in such a structure, one can look at the distributions with the "full" scale. One will notice that modes 1 and 3 obviously have the same "even" symmetry in the E fields, while mode 2 has an odd symmetry. The CPW mode has an "even" symmetry, so it has to be mode 1 or 3.

Modes 1 and 2 have significant E-field strengths in the substrate, especially under the "ground" strips. So, there is a potential difference between the "ground" strips and the outer ground for these two modes. However, for mode 3, the "ground" strips are at the same potential as the outer ground, which is consistent with the CPW mode.



Thus one can identify the modes. For mode 3, the "ground" strips are at 0 volts with respect to the outer ground, and the signal line has +V. The  $Z_{pv}$  impedance computed for this mode using a dot product of the voltage vector for mode 3 gives the same  $Z_{pv}$  as by computing the simple voltage between center strip and either "ground" strip. That is because the

voltage vector for mode 3 along the three impedance lines is  $\bar{V} = [0, V, 0]$  and

$$\bar{V} \cdot \bar{V} = [0, V, 0] \cdot [0, V, 0] = 0 \cdot 0 + V^2 + 0 \cdot 0 = V^2$$

However, the impedances for the other modes now match accepted impedance definitions found in the literature for multi-conductor transmission lines.

## Application Areas

EM modeling tools are known for their great accuracy. FEM Simulator redefines this term with broad application coverage, including the following:

- Microstrip, stripline, CPW elements (filters, couplers, spiral inductors, via holes, air bridges, meander lines...)
- Multilayer structures
- Ceramic filters
- Adapters/transitions
- Antennas
- Couplers
- Power splitters/combiners

## Visualization and Display of Results

The visualization and animation capabilities in FEM Simulator enable you to evaluate simulation results thoroughly. For analyzing designs, you can perform the EM field animation and dynamic rotation of structures simultaneously. You can choose from shaded plots or vectors. 3D far-field plots illustrate beam shapes in both azimuth and elevation on a single plot. To aid in analyzing your designs, EM field animation and dynamic rotation of structures can be performed simultaneously. You can choose from shaded plots, contour lines, or vectors. 3D far-field plots illustrate beam shapes in both azimuth and elevation on a single plot.

## FEM Simulator Process Overview

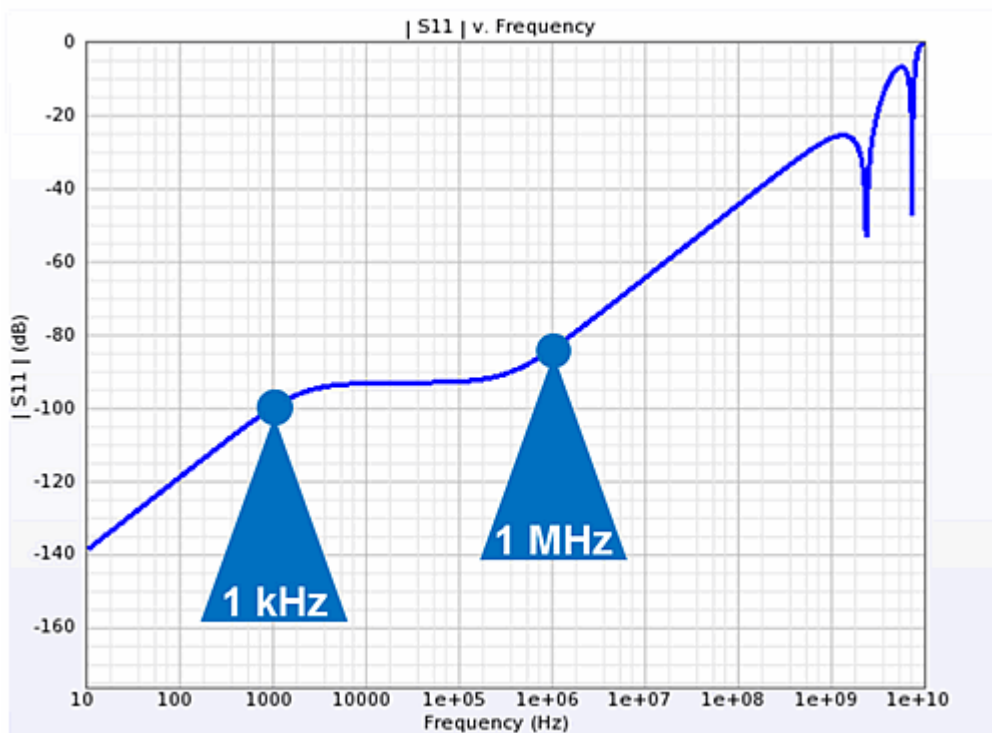
The FEM Simulator commands are available from the EMPro GUI. The following steps describe a typical process for creating and simulating a design with the FEM Simulator:

1. **Create a physical design:** Create a physical design for your FEM simulation in EMPro.
2. **Set Simulation Options:** A mesh is a pattern of tetrahedra that is applied to a design in order to break down (discretize) the design into small cells. A mesh is required in order to simulate the design effectively. You can specify a variety of mesh parameters to customize the mesh to your design, or use default values and let FEM generate an optimal mesh automatically.

3. **Simulate the circuit:** You set up a simulation by specifying the parameters of a frequency plan, such as the frequency range of the simulation and the sweep type. When the setup is complete, you run the simulation. The simulation process uses the mesh pattern, and the electric fields in the design are calculated. S-parameters are then computed based on the electric fields. If the Adaptive Frequency Sample sweep type is chosen, a fast, accurate simulation is generated, based on a rational fit model.
4. **View the results:** The data from an FEM simulation is saved as S-parameters or as fields.
5. **FEM Advance Visualization:** The FEM Advance Visualization tool enables you to view and analyze, S-parameters, far-fields, and antenna parameters. Data can be analyzed in a variety of 2D and 3D plot formats. Some types of data are displayed in tabular form.
6. **Radiation patterns:** Once the electric fields on the circuit are known, the electromagnetic fields can be computed. They can be expressed in the spherical coordinate system attached to your circuit.

## New FEM Low Frequency Algorithm

FEM is the most popular method for 3D EM simulation in frequency domain. However, the traditional full-wave FEM solver has low frequency stability issues. The system matrix of the solver is not suitable at low frequencies. As a result, the accuracy of the solver deteriorates below MHz range. If the mesh element is small, the solver becomes unstable even at higher frequencies. New technologies are applied in the FEM solver to enhance the low frequency stability. With these enhancements, accurate and stable results can be achieved from microwave frequency down to DC.



The new FEM DC/LF algorithm provides following benefits:

- Enables stable, accurate results down to very low frequencies.
- Provides accurate DC results for use in circuit simulation.
- Early benchmarks show superior accuracy relative to alternative EM solvers.

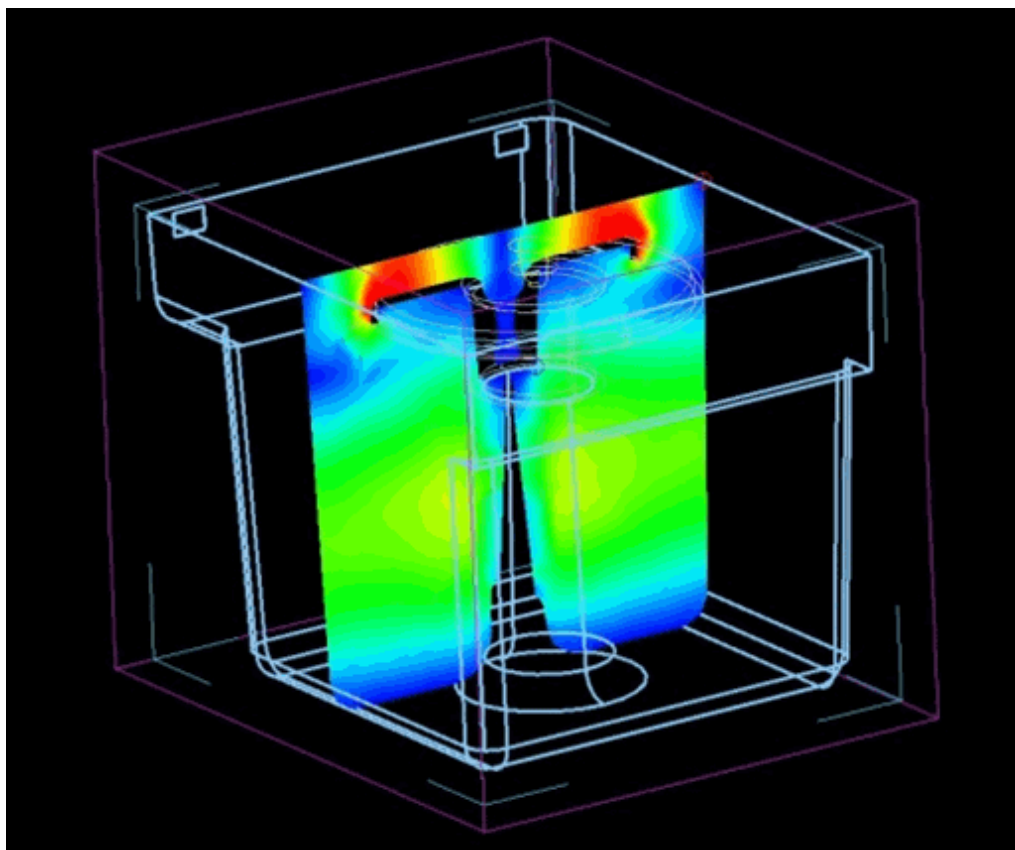
Accurate DC and low frequency EM simulation are significant in RF/microwave and high-speed digital (HSD) designs. An accurate and efficient low frequency solution is also required to capture resonances at very low frequencies.

## Eigenmode Overview

You can use an Eigenmode solver to generate resonance properties (eigenmodes) of a closed structure without enforcing excitations. In the results, you can view the Eigen frequencies, Q value, Eigen field, and surface currents at each Eigen mode. This new solver quickly finds the resonant frequencies for devices such as cavity filters, which is a common high-frequency component used in wireless communication systems. Filter designers can also visualize the resulting electromagnetic fields at each resonant frequency and make adjustments to the cavity structure to optimize filter performance.

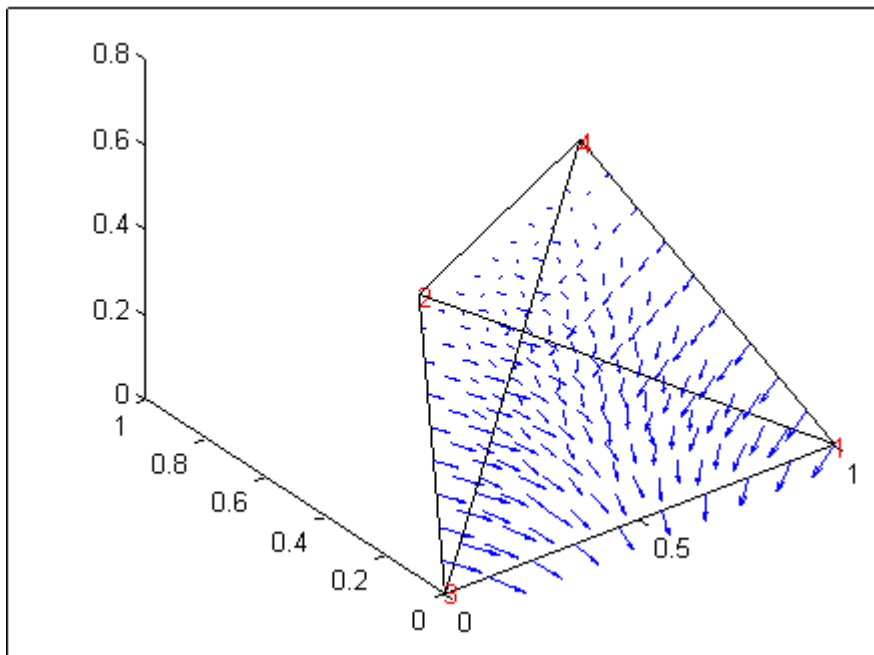
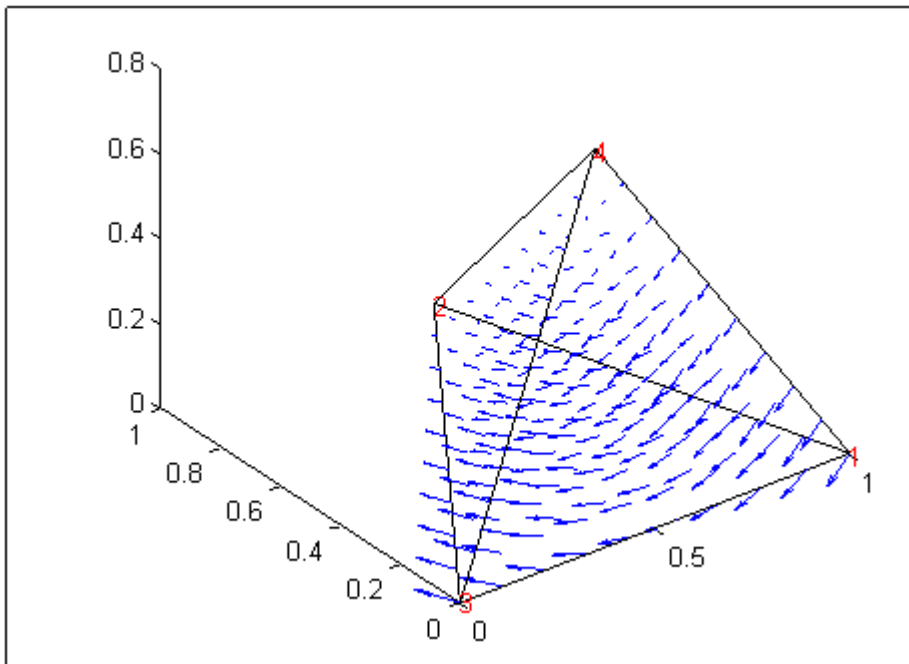
The Eigenmode solver is based on FEM technology and you need an FEM Simulator Element license. The existing EMPro FEM solver, regular mode solver, is driven by excitations and generates S parameters and/or radiation fields. The Eigenmode solver process is similar to a typical FEM or FDTD flow in 3D EM simulations.

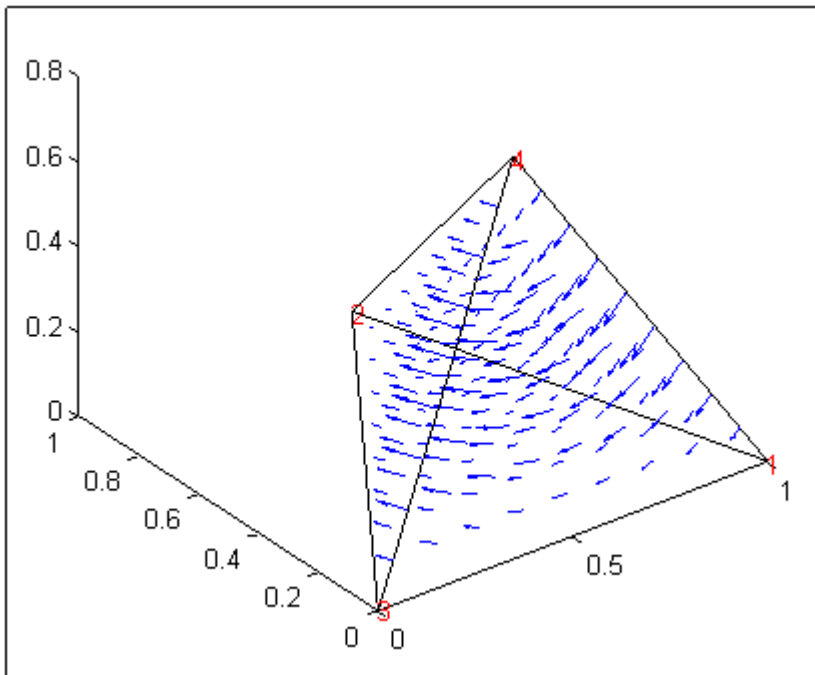
The following figure displays the visualization results of an Eigenmode simulation:



### Tetrahedron sampling

Eigenmode solver in EMPro is based on FEM technology, where the solution space is subdivided to a number of mesh cells. Each mesh cell is a tetrahedron. 1st and 2nd order basis functions are built to represent the field in each cell in following figure:





## Materials and Boundary Conditions

The EMPro eigenmode simulation supports all materials that are supported by FEM simulation. It also supports all the boundary conditions that are supported by FEM simulation, except for the radiation boundary conditions.

## Eigenmode Matrix System

Due to the interaction between the basis functions with the material properties and boundary conditions, the FEM matrix  $M(f)$  can be generated. Elements in  $M$  can be a non-linear function of the frequency  $f$ . The eigenmode solver is to find the eigenvalues (eigenfrequencies) and the corresponding eigen vectors (eigenfields) directly without exciting the system:

$$\mathbf{M}(f_i) \cdot \mathbf{x}_i = 0,$$

Where  $f_i$  is the  $i$ th eigenfrequency and  $x_i$  is the  $i$ th eigenfield. Note that  $f_i$  can be a complex number where the real part represents the resonant frequency and the imaginary part is related to the quality factor of the resonator.

## Mesh Accuracy and Refinement

The Eigenmode simulator implements an adaptive mesh algorithm, where an initial mesh is generated and the electric fields (and eigen frequencies) are computed on that initial mesh for a single frequency. An error estimate is generated for each tetrahedron. The tetrahedra with the largest estimated error are refined to create a new mesh on which the electric fields eigen frequencies are computed. The eigen frequencies from consecutive meshes are compared. If the eigen frequencies do not change significantly, then electric fields are computed for all the requested frequencies. However, if the frequencies do change significantly, then new error estimates are computed, a new mesh is generated and new electric fields and frequencies are computed.

## Result Display

The results of the eigenmode solver include the eigenvalues (eigenfrequencies and quality factors) and eigenvectors (eigenfields). The eigenvalues are printed in the EMPro Simulation log window as shown in the following figure. They also available from the Results window.

	Eigenfrequency	Q value
1.	400.816 MHz	1.316900e+003
2.	3.04314 GHz	3.317712e+003
3.	5.12743 GHz	5.969380e+002
4.	5.12743 GHz	5.969380e+002
5.	6.00024 GHz	4.642562e+003

The eigen fields are available in the Advanced Visualization and EMPro Results window.

## FEM 2D Port Solver Overview

Before you can calculate the three dimensional electromagnetic field inside a structure, it is necessary to determine the excitation field pattern at each port. The FEM 2D solver calculates the natural field patterns (or modes) that can exist inside a transmission structure with the same cross section as the port. The resulting 2D field patterns serve as boundary conditions for the full three- dimensional problem.

### Excitation Fields

The assumption is that each port is connected to a uniform waveguide that has the same cross section as the port. The port interface is assumed to lie on the  $z=0$  plane. Therefore, the excitation field is the field associated with traveling waves propagating along the waveguide to which the port is connected:

$$E(x,y,z,t) = \text{Re}[E(x,y)e^{j\omega t - \gamma z}]$$

where:

- Re is the real part of a complex number or function.
- $E(x,y)$  is a phasor field quantity.
- $j$  is the imaginary unit, .
- $\omega$  is angular frequency,  $2\pi f$ .
- $\gamma = a + jb$  is the complex propagation constant,
- $a$  is the attenuation constant of the wave.
- $b$  is the propagation constant associated with the wave that determines, at a given time  $t$ , how the phase angle varies with  $z$ .

In this context, the  $x$  and  $y$  axes are assumed to lie in the cross section of the port; the  $z$  axis lies along the direction of propagation.

## Wave Equation

The field pattern of a traveling wave inside a waveguide can be determined by solving Maxwell equations. The following equation that is solved by the 2D solver is derived directly from Maxwell equation.

$$\nabla \times \left( \frac{1}{\mu_r} \nabla \times \mathbf{E}(x,y) \right) - k_0^2 \epsilon_r \mathbf{E}(x,y) = 0$$

where:

- $\mathbf{E}(x,y)$  is a phasor representing an oscillating electric field.
- $k_0$  is the free space wave number, ,
- $\omega$  is the angular frequency, 2 $\pi$ f.
- $\mu_r(x, y)$  is the complex relative permeability.
- $\epsilon_r(x, y)$  is the complex relative permittivity.

To solve this equation, the 2D solver obtains an excitation field pattern in the form of a phasor solution,  $\mathbf{E}(x,y)$ . These phasor solutions are independent of  $z$  and  $t$ ; only after being multiplied by  $e^{-j\omega z}$  do they become traveling waves. Also note that the excitation field pattern computed is valid only at a single frequency. A different excitation field pattern is computed for each frequency point of interest.

## Modes

For a waveguide or transmission line with a given cross section, there is a series of basic field patterns (modes) that satisfy Maxwell's equations at a specific frequency. Any linear combination of these modes can exist in the waveguide.

## Mode Conversion

In some cases it is necessary to include the effects of higher- order modes because the structure acts as a mode converter. For example, if the mode 1 (dominant) field at one port is converted (as it passes through a structure) to a mode 2 field pattern at another, then it is necessary to obtain the S- parameters for the mode 2 field.

## Modes, Reflections, and Propagation

It is also possible for a 3D field solution generated by an excitation signal of one specific mode to contain reflections of higher- order modes which arise due to discontinuities in a high frequency structure. If these higher- order modes are reflected back to the excitation port or transmitted onto another port, the S- parameters associated with these modes should be calculated.

If the higher- order mode decays before reaching any port—either because of attenuation due to losses or because it is a non- propagating evanescent mode—there is no need to obtain the S- parameters for that mode. Therefore, one way to avoid the need for computing the S- parameters for a higher- order mode is to include a length of waveguide in the geometric model that is long enough for the higher- order mode to decay.

For example, if the mode 2 wave associated with a certain port decays to near zero in 0.5 mm, then the “constant cross section” portion of the geometric model leading up to the port should be at least 0.5 mm long. Otherwise, for accurate S-parameters, the mode 2 S- parameters must be included in the S-matrix.

The length of the constant cross section segment to be included in the model depends on the value of the mode's attenuation constant,  $\alpha$ .

## Modes and Frequency

The field patterns associated with each mode generally vary with frequency. However, the propagation constants and impedances always vary with frequency. Therefore, when a frequency sweep has been requested, a solution is calculated for each frequency point of interest. When performing frequency sweeps, be aware that as the frequency increases, the likelihood of higher- order modes propagating also increases.

**Modes and Multiple Ports on a Face** Visualize a port face on a microstrip that contains two conducting strips side by side as two separate ports. If the two ports are defined as being separate, they are treated as two ports are connected to uncoupled transmission structures. It is as if a conductive wall separates the excitation waves. However, in actuality, there will be electromagnetic coupling between the two strips. The accurate way to model this coupling is to analyze the two ports as a single port with multiple modes.

## Displaying Fields by Mode

After a port solution has been generated, an arrow plot of the E- field distribution can be viewed. If more than one mode is being analyzed at that port, the field patterns associated with each mode can be displayed.



# Specifying FEM Simulation Setup

---


Before running an FEM simulation, you need to specify the simulation options that are specific to the FEM simulator. You can specify the simulation options that are specific to the FEM Simulator in the Setup FEM Simulation window. To run an FEM simulation, you need to perform the following tasks:

- Selecting the FEM simulator.
- Reusing simulation results.
- Specifying frequency plans.
- Specifying mesh refinement settings.
- Adding Notes about your FEM simulation.
- Completing the setting up process.
- Running the FEM simulation.

## Selecting the FEM Simulator

---

You can specify the simulation options that are specific to the FEM simulator in the Setup FEM Simulation window. To open the Setup FEM Simulation window:

1. Select **FEM** from the drop-down list available in the **Simulation** toolbar.
2. Click **Setup** (  ) to create and edit the setup of a new simulation. The Setup FEM Simulation window is displayed.
3. Type a name for your FEM simulation name in the **Name** field. You can also use the default name that is specified in the Setup FEM Simulation window. The invalid symbol on the window refers that you have not entered any frequency plan or boundary conditions are not correct for FEM simulations.

## Reusing the Simulation Results

---

EMPro allows you to reuse any existing FEM simulation. For a new FEM simulation, simulation setup in terms of mesh refinement, AFS points, solver can be reused from existing simulation. In Setup FEM Simulation window, the **Simulation Results to Reuse** drop-down list displays the list of already solved simulations. You can choose the FEM simulation to be reused using the **Simulation Results to Reuse** drop-down list.

To know more about the instances which are useful to reuse, see [Reusing Mesh and Frequency Points](#) section.

## Defining FEM Simulation Options

---

- [Specifying Frequency Plans in FEM](#)
- [Specifying Mesh and Refinement Properties in FEM](#)

- **Selecting a Solver in FEM**

**NOTE**

If you want to add any notes or observation with your simulation, you can specify it in the Notes text box. Click **Notes** in the Setup FEM Simulation window to display the Notes screen.

## Completing the Process of Specifying Setup

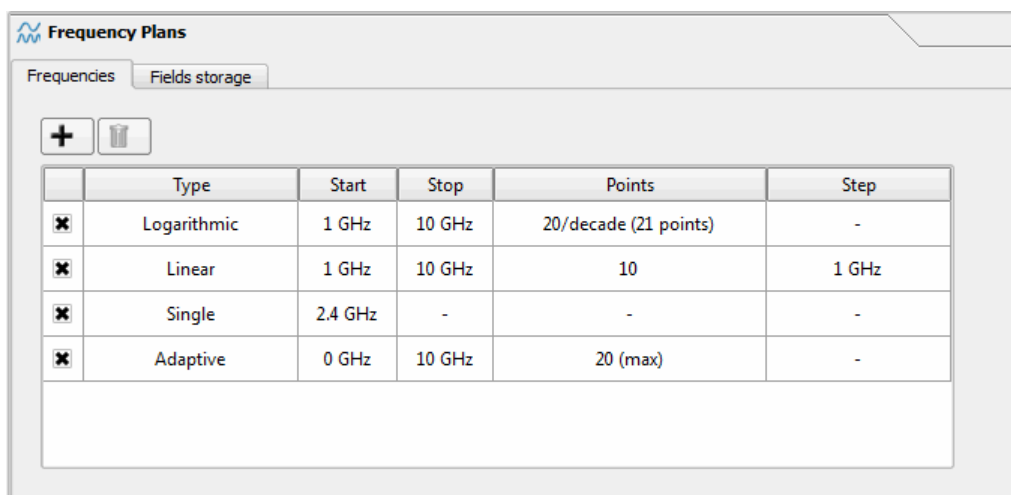
After you have completed entering your FEM Simulation options, click the **Done** button to apply the current settings in the Setup FEM Simulation window, or click **Create Simulation Only** to accept the settings. You can also click **Create and Queue Simulation** to create and queue the simulation. Alternatively, you can click the **Cancel** button to abort the changes and dismiss the dialog box.

The Simulations window stores the project simulations. You can create, queue, and run simulations using the Simulations workspace.

## Specifying Frequency Plans in FEM

You can set up multiple frequency plans for an FEM simulation. For each plan, you can specify that a solution be found for a single frequency point or over a frequency range. You can also select a sweep type for your frequency plan. However, the collection of frequency plans run as a single simulation. You can specify the frequency settings for your FEM simulation in the Setup FEM Simulation window. In addition, you can save results for all frequencies or user-defined frequency.

The following figure displays the Frequency Plans tab:



You can specify the following options for creating a frequency plan:

### Sweep Type

Generally, sweeps of individual parameters can be performed most efficiently from within many of the simulator dialog boxes themselves. The ability to step through a series of values automatically is incorporated into all the standard simulation controllers.

- **Adaptive:** It is the preferred sweep type. It uses a fast and accurate method of comparing sampled S-parameter data points to a rational fitting model. The value entered in the Sample Points Limit field is the maximum number of samples used in an attempt to achieve convergence. The solutions from the final attempt will be saved. If convergence is achieved using fewer samples, the solutions are saved and the simulation will end.
- **Linear:** It simulates over a frequency range, selecting the frequency points to be simulated in linear increments based on the step size you specify. Type the start and the stop frequencies in the Start and Stop fields, and select frequency units for each. Enter the step size in the Step field, and select units.
- **Log:** It simulates over a frequency range, selecting the frequency points to be simulated in logarithmic increments. Type the start and the stop frequencies in the Start and Stop fields, and select frequency units for each. Enter the number of frequency points to be simulated per decade of frequency in the Points/Decade field, and select units.
- **Single:** It simulates at a single frequency point. Type the value in the Frequency field and select the units.

## Start Frequency

Specify the start frequency value in GHz.

## Sample Points Limit


Specifies the number of frequencies to simulate. For Adaptive, this is the maximum number of samples to use.

## Stop Frequency

Specify the stop frequency value in GHz.

## Defining a Frequency Plan

To add a frequency plan in the Setup FEM Simulation window:

1. Open the **Setup FEM Simulation** window.
2. Click  to insert a new row.
3. Click once in the **Type** field and select a sweep type from the drop-down list.
4. Type a value in the **Start Frequency** text box or you can accept the default **minFreq** value.
5. Type a value in the **Stop Frequency** text box or you can accept the default **maxFreq** value.

### NOTE

You can modify a frequency plan by clicking **Update**. To delete a frequency plan, click **Delete**.

## Selecting a Field Storage Option

When you run an FEM simulation, the results can be saved for all frequencies from the frequency plan or user-defined frequencies.

The following table describes the fields storage options:

Option	Description
All frequencies from the frequency plan and mesh frequency	Field data will be stored for the frequencies for which either mesh is refined or solution is calculated.
No field data	Field data will not stored for any of the frequencies used in calculation. This will save space on hard disk. If the intention is just to see S parameter, this option can be used.
User defined frequencies	Field data will be stored for only specified frequencies. The specified frequencies are the frequencies explicitly specified in the frequency plan, i.e. the start and end frequencies of an Adaptive sweep, any single frequency, all the frequencies of the linear & logarithmic frequencies. Field solutions will <b>NOT</b> be saved for frequency automatically determined to sample the Adaptive frequency sweep.

## Specifying Mesh and Refinement Properties in FEM

An FEM simulation mesh is a sub-division of the entire 3D problem domain into a set of tetrahedra (or cells). This pattern of cells is based on the geometry and optionally on user-defined parameters, so each geometry will have a unique mesh calculated for it. The mesh is then applied to the circuit in order to compute the electromagnetic fields within each cell and identify any coupling effects in the geometry during simulation. From these calculations, S-parameters are then calculated for the circuit.

The FEM Simulator implements an adaptive mesh algorithm, where an initial mesh is generated and the electromagnetic fields (and S-parameters) are computed on that initial mesh for a single frequency (or multiple frequencies if broadband refinement is used). An error estimate is generated for each tetrahedron. The tetrahedra with the largest estimated error are refined to create a new mesh on which the electromagnetic fields and S-parameters are computed. The S-parameters from consecutive meshes are compared. If the S-parameters do not change significantly, then electromagnetic fields (and S-parameters) are computed for all the requested frequencies. If the S-parameters do change significantly, then new error estimates are computed, a new mesh is generated and new fields (and S-parameters) are computed.

Click **Mesh/Refinement Properties- (Delta Error: 0.01)** in the Setup FEM Simulation window. This displays the **Mesh/Refinement Properties- (Delta Error: 0.01)** screen, which consists of four tabs: **Stop criterium**, **Refinement**, **Initial mesh** and **Advanced**.

## Stop Criterium Tab Settings

The screenshot shows the 'Mesh/Refinement Properties - (Delta Error: 0.01)' dialog box with the 'Stop criterium' tab selected. The settings are as follows:

- Delta error:** 0.01
- Consecutive passes of delta error required:** 1
- Minimum number of adaptive passes:** 1
- Maximum number of adaptive passes:** 15
- Memory limit:** 1.9 GB
- ☒ **Limit memory**

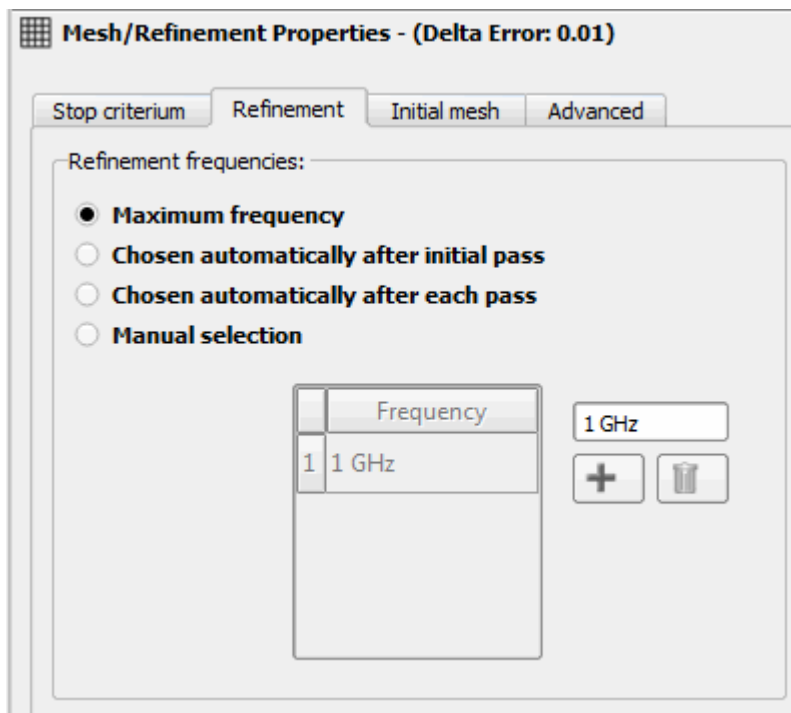
The following table describes the **Stop criterium** tab settings:

Option	Description
Delta Error	Global Delta S-parameter sets a value that is applied to all S-parameters in the solution. Enter a value in the Delta Error field. This is the allowable change in the magnitude of the vector difference for all S-parameters for at least the number of consecutive refinement passes as specified in the field "Consecutive passes of delta error required".
Consecutive passes of delta error required	Specify the required number of consecutive passes with a delta error smaller than or equal to the value specified in the field "Delta Error".
Minimum number of adaptive passes	Enter the minimum number of refinement passes. The number of refinement passes will be at least this number even if the convergence criterium based on the delta error is met with less refinement passes. If you have concerns about convergence, you can increase this value, otherwise, use the default value of 2.
Maximum number of adaptive passes	Enter the maximum number of passes to be attempted. If the number of refinement passes entered is reached before the delta error criterium is met, the refinement process will end, based upon this limit. Typically, a value between 10 and 20 is recommended.
Memory Limit	

Option	Description
	<p>If the memory used by the solver exceeds this memory limit, the adaptive mesh refinement will end and the simulation will continue with the solve phase, even if the delta error does not satisfy the convergence criterium.</p> <p>Select the <b>Limit Memory</b> check box to enable this feature.</p>

## Refinement Tab Settings

Using the **Refinement** tab settings you can control the frequencies at which the mesh will be adaptively refined.



The following table describes the **Refinement** tab settings:

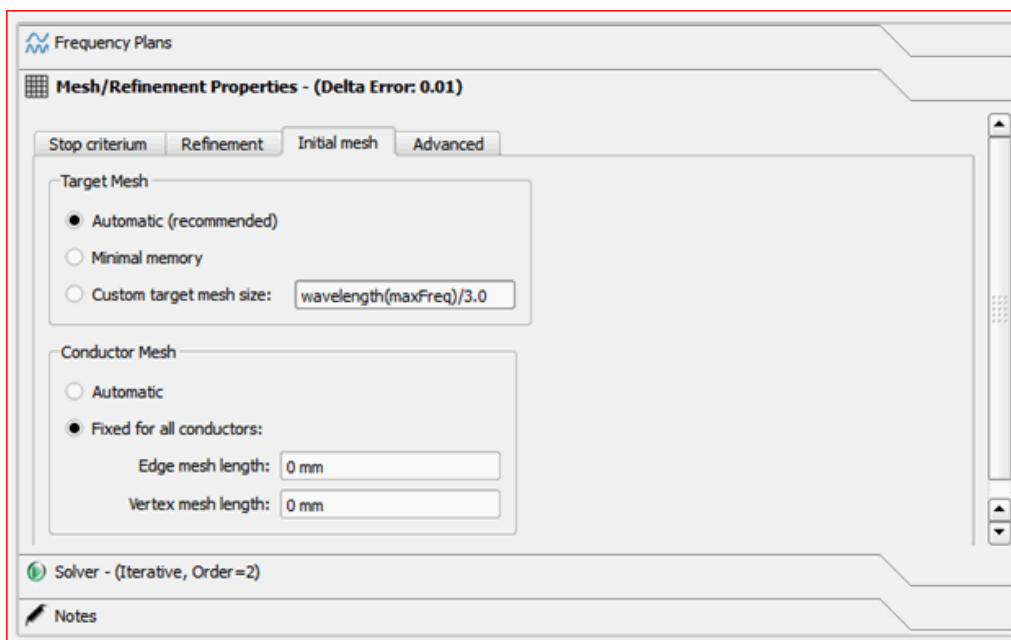
Option	Description
Maximum frequency	The refinement frequency will be the highest frequency found in the frequency plan.
Chosen automatically after initial pass	Refinement frequencies are chosen by the solver. This strategy is recommended when you are not aware about the important characteristics that can occur in your frequency, but you know the frequency transitions are not extremely sharp (i.e. are not high Q resonances).
Chosen automatically after each pass	

Option	Description
	<p>Refinement frequencies are not fixed, but can vary from level to level. This is an automated option and tends to generate accurate results, but it also tends to result in the longest simulation times. This strategy is recommended:</p> <ul style="list-style-type: none"> <li>• For structures with high Q resonances.</li> <li>• During final analysis of a structure where accuracy is preferred over fast simulations.</li> </ul>
Manual Selection	Refinement frequencies are specified manually.

For more information, see [Performing FEM Broadband Refinement](#).

## Initial Mesh Tab Settings

The initial mesh is the starting point for the mesh refinement process, part of the FEM simulation. The Initial mesh settings allow you to control the first mesh.



The following table describes the Target Mesh and Conductor Mesh options:

Option	Description
Target Mesh	The Target Mesh options specifies the size of edges on tetrahedra:

Option	Description
	<ul style="list-style-type: none"> <li>Automatic: Automatic is the recommended setting. The target mesh length will be one third of the free space wavelength at the maximum frequency. It is a compromise between accuracy and simulation performance.</li> <li>Minimal memory: When Minimal Memory is chosen, the initial mesh will generate a mesh with the least amount of tetrahedra as possible.</li> <li>Custom target mesh size: The custom target mesh works similar to the automatic target mesh except that as a user you have the liberty to specify the mesh size.</li> </ul>
Conductor mesh	<ul style="list-style-type: none"> <li>Automatic: If you select the Automatic conductor mesh option, during the simulation process, the DC connectivity of the design is extracted and all conductive objects connected to a port are considered.</li> <li>Fixed for all Conductors: All conductors receive the specified mesh length.</li> </ul>

For more information about the initial mesh options, see [Initial Mesh Settings](#).

#### See Also

[Initial Mesh Settings](#)

[Guidelines for Specifying Mesh Settings](#)

## Specifying Advanced Tab Settings

Click the **Advanced** tab to display the advanced options:

The screenshot shows the 'Advanced' tab selected in a software interface. The 'Target Mesh Growth' is set to 25. The 'Use Initial Minimal Mesh Size' checkbox is checked, with a value of 0.3 mm and an 'Automatically determine' button. The 'Merge Objects with Same Material' checkbox is unchecked. The 'After convergence, solve on the finest mesh' checkbox is checked. Under 'Automatic conductor mesh settings', both 'Edge meshing' and 'Vertex meshing' are checked. 'Edge meshing' is set to 0.2 x estimated conductor width, and 'Vertex meshing' is set to 0.2\*0.3 x estimated conductor width.

You can specify the following advanced options:



Option	Description
Target Mesh Growth	Specifies the percentage by which each adaptive mesh grows relative to the previous mesh.
Use Initial Mesh Size	Enables you to set a minimum length of edges in the initial mesh. During mesh refinement, the individual length of edges in the mesh can be shortened where deemed necessary by the adaptive refinement process. If you click <b>Automatically Determined</b> , a value is automatically calculated based on the requested frequencies and geometric size. Using this option, you can restrict the meshing process to a more limited set of possible solutions. It is advised to only use this options when the initial mesh is unreasonably dense in some areas due to some combination of geometric features triggering that behavior.
Merge Objects with Same Material	This option controls if two objects that have the same material properties may be merged prior to meshing. This option can be used to reduce the number of tetrahedrons generated. Object merging is applied only when the objects also have the same mesh priority.
After convergence, solve on the finest mesh	Performs simulation on the finest mesh after convergence.
Automatic conductor mesh setting Edge meshing	This option is used when automatic meshing is applied. It determines the length of the edge mesh in relation to the estimated width of the conductor.
Automatic conductor mesh setting: Vertex meshing	This option is used when automatic meshing is applied. It determines the length of the vertex mesh in relation to the estimated width of the conductor.

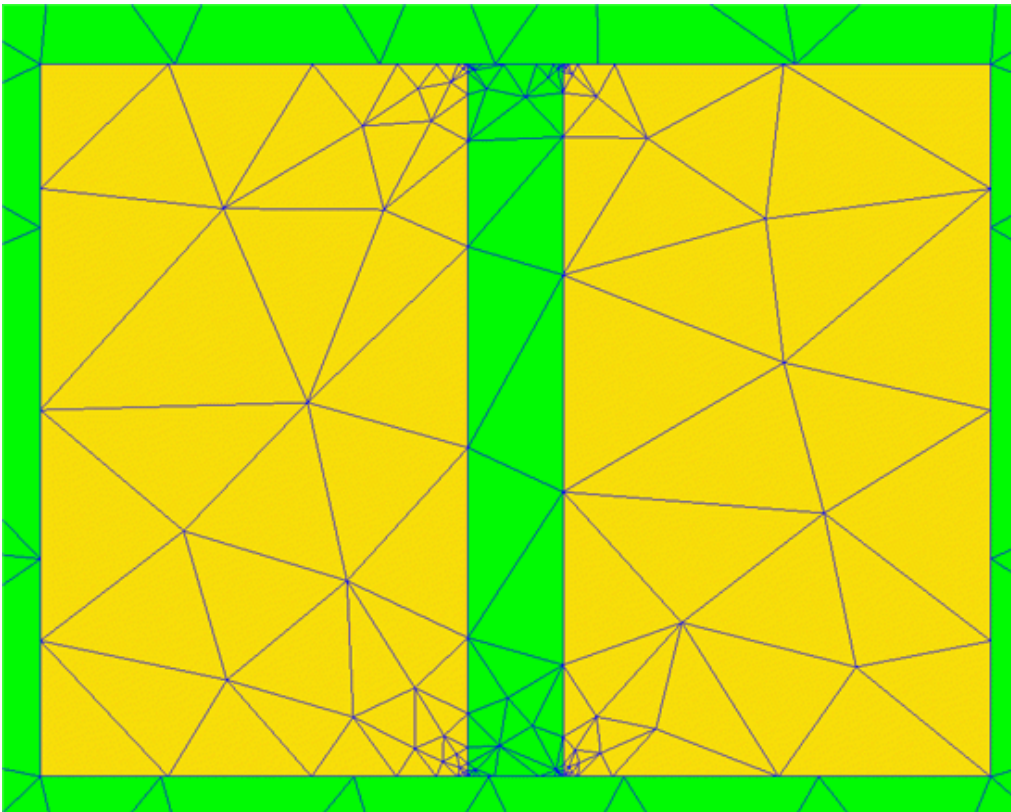
**NOTE**

Advanced meshing options should be changed only in case of meshing problems. For more information about advanced meshing options, see [Initial Mesh Settings](#)

## Vertex Mesh

Vertex meshing reduces the mesh length around vertices. When combined with edge meshing the shortest length of both is chosen in any given region. The region where the value is applied by the vertex mesh length setting is roughly twice its length in each direction: the actual region is still being optimized. The region affected by it depends on local geometry.

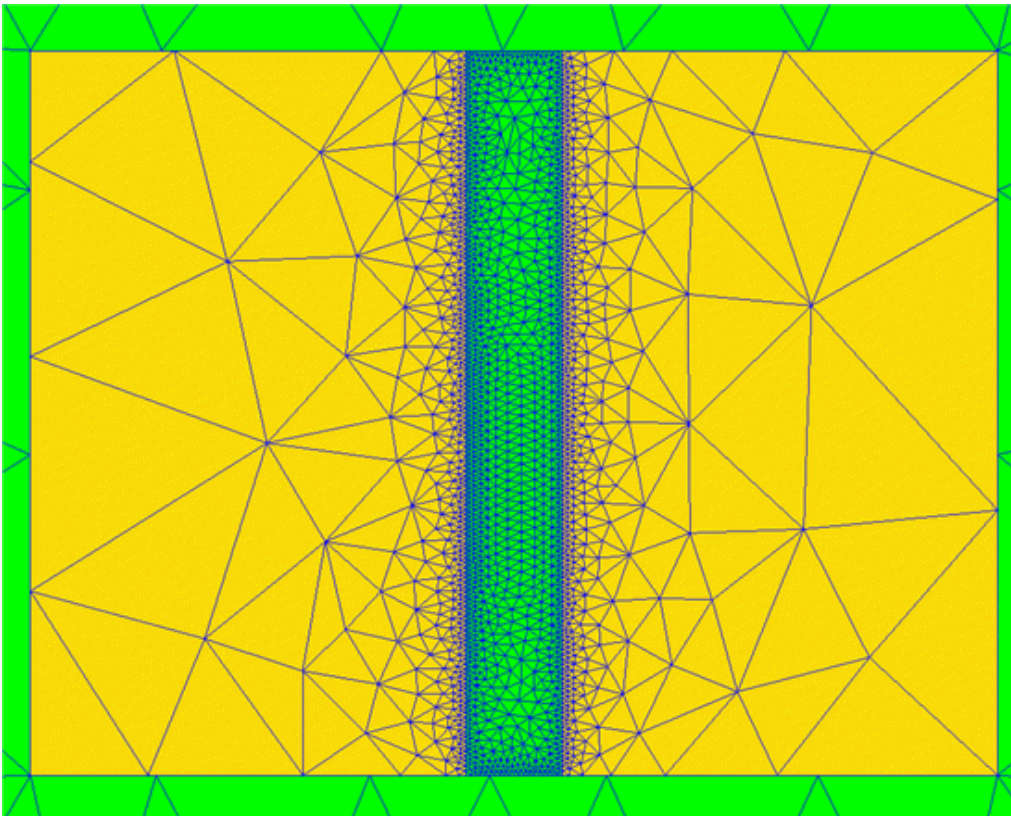
The following figure displays the result of a vertex mesh setting:



## Edge Mesh

Edge meshing reduces the mesh length around the edges. A fan-out effect is created to maintain the tetrahedron quality. The edges do not need to be straight lines, any shape is allowed.

The following figure displays the result of an edge mesh setting:



# Specifying Mesh Refinement Properties in FEM

An FEM simulation mesh is a sub-division of the entire 3D problem domain into a set of tetrahedra (or cells). This pattern of cells is based on the geometry and optionally on user-defined parameters, so each geometry will have a unique mesh calculated for it. The mesh is then applied to the circuit in order to compute the electromagnetic fields within each cell and identify any coupling effects in the geometry during simulation. From these calculations, S-parameters are then calculated for the circuit.

The FEM Simulator implements an adaptive mesh algorithm, where an initial mesh is generated and the electromagnetic fields (and S-parameters) are computed on that initial mesh for a single frequency (or multiple frequencies if broadband refinement is used). An error estimate is generated for each tetrahedron. The tetrahedra with the largest estimated error are refined to create a new mesh on which the electromagnetic fields and S-parameters are computed. The S-parameters from consecutive meshes are compared. If the S-parameters do not change significantly, then electromagnetic fields (and S-parameters) are computed for all the requested frequencies. If the S-parameters do change significantly, then new error estimates are computed, a new mesh is generated and new fields (and S-parameters) are computed.

Click **Mesh/Refinement Properties- (Delta Error: 0.01)** in the Setup FEM Simulation window. This displays the **Mesh/Refinement Properties- (Delta Error: 0.01)** screen, which consists of four tabs: **Stop criterium**, **Refinement**, **Initial mesh** and **Advanced**.

## Stop Criterium Tab Settings

The screenshot shows the 'Mesh/Refinement Properties - (Delta Error: 0.01)' dialog box with the 'Stop criterium' tab selected. The dialog has four tabs: 'Stop criterium', 'Refinement', 'Initial mesh', and 'Advanced'. The 'Stop criterium' tab contains the following settings:

- Delta error:** 0.01
- Consecutive passes of delta error required:** 1
- Minimum number of adaptive passes:** 1
- Maximum number of adaptive passes:** 15
- Memory limit:** 1.9 GB
- ☒ **Limit memory**

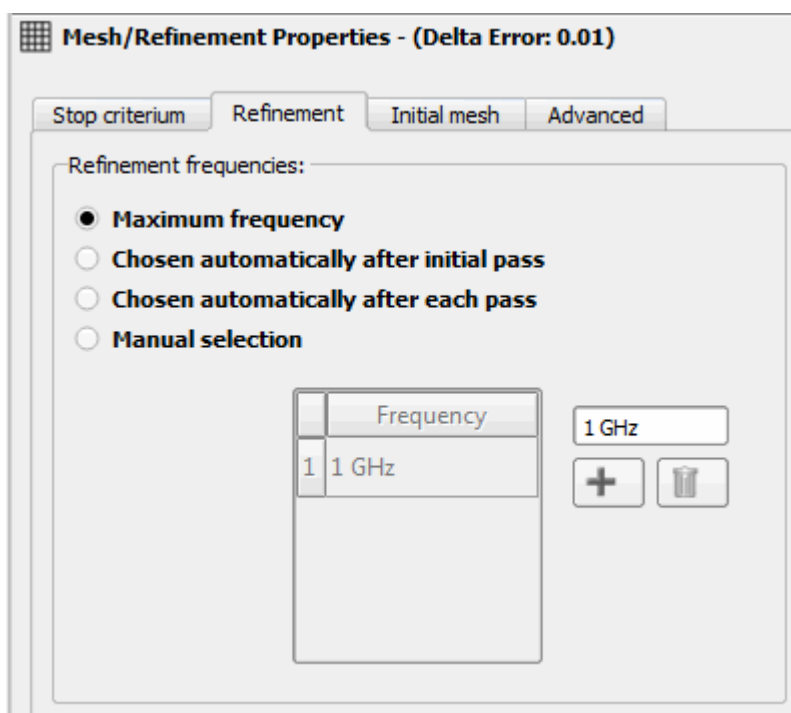
The following table describes the **Stop criterium** tab settings:

Option	Description
Delta Error	Global Delta S-parameter sets a value that is applied to all S-parameters in the solution. Enter a value in the Delta Error field. This is the allowable change in the magnitude of the vector difference for all S-parameters for at least the number of consecutive refinement passes as specified in the field "Consecutive passes of delta error required".
Consecutive passes of delta error required	

Option	Description
	Specify the required number of consecutive passes with a delta error smaller than or equal to the value specified in the field "Delta Error".
Minimum number of adaptive passes	Enter the minimum number of refinement passes. The number of refinement passes will be at least this number even if the convergence criterium based on the delta error is met with less refinement passes. If you have concerns about convergence, you can increase this value, otherwise, use the default value of 2.
Maximum number of adaptive passes	Enter the maximum number of passes to be attempted. If the number of refinement passes entered is reached before the delta error criterium is met, the refinement process will end, based upon this limit. Typically, a value between 10 and 20 is recommended.
Memory Limit	If the memory used by the solver exceeds this memory limit, the adaptive mesh refinement will end and the simulation will continue with the solve phase, even if the delta error does not satisfy the convergence criterium. Select the <b>Limit Memory</b> check box to enable this feature.

## Refinement Tab Settings

Using the **Refinement** tab settings you can control the frequencies at which the mesh will be adaptively refined.



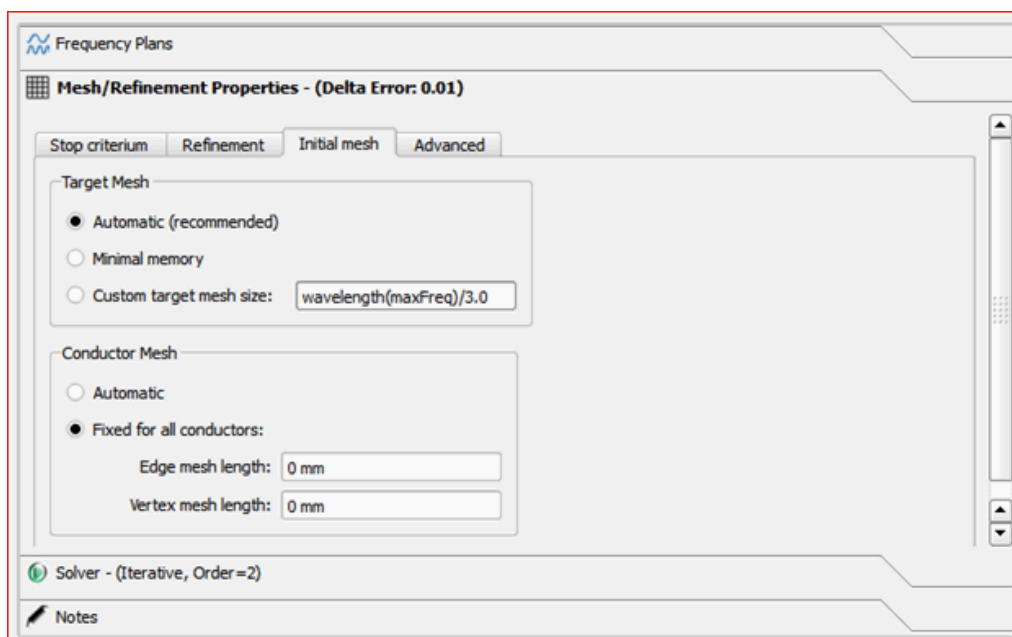
The following table describes the **Refinement** tab settings:

Option	Description
Maximum frequency	The refinement frequency will be the highest frequency found in the frequency plan.
Chosen automatically after initial pass	Refinement frequencies are chosen by the solver. This strategy is recommended when you are not aware about the important characteristics that can occur in your frequency, but you know the frequency transitions are not extremely sharp (i.e. are not high Q resonances).
Chosen automatically after each pass	Refinement frequencies are not fixed, but can vary from level to level. This is an automated option and tends to generate accurate results, but it also tends to result in the longest simulation times. This strategy is recommended: <ul style="list-style-type: none"> <li>• For structures with high Q resonances.</li> <li>• During final analysis of a structure where accuracy is preferred over fast simulations.</li> </ul>
Manual Selection	Refinement frequencies are specified manually.

For more information, see [Performing FEM Broadband Refinement](#).

## Initial Mesh Tab Settings

Using the **Initial mesh** tab settings you can control how the initial mesh is generated.



## Background Mesh options

- Automatically: EMPro automatically determines the initial target mesh size. The length is computed based on the frequency plan available when the request is made.
- Minimal Memory: EMPro uses minimum memory for calculating the mesh.
- Custom Target mesh size: Provides an initial denser mesh. If used, no edge in the initial mesh will be longer than the given length.

### Conductor Mesh Options

- Automatic: When automatic conductor meshing is selected, the mesher estimates where mesh lengths need to be reduced to improve the accuracy and convergence speed of the simulation. The algorithm to produce this mesh is based on the DC connectivity of geometry with ports. Only parts electrically connected to a port are affected. For those parts the edge and vertex mesh is set. The length of the edge and vertex mesh is based on the estimated width assuming the structure being excited starts with a strip-like construction. The mesh process will report the estimated width in the simulation output window. |

The following options are used for specifying the mesh for all conductors:

- Edge mesh length: Specifies a target mesh size of all edges that belong to the geometry of a conductor. Edges do not necessarily have to be straight lines. A value of 0 means that no initial conductor edge mesh length is applied.
- Vertex mesh length: Specifies a target mesh size for all vertices that belong to the geometry of a conductor. A value of 0 means that no initial conductor vertex mesh length is applied.

For all lengths a parameter or formula is allowed. The formula is always evaluated at the time the simulation is being created.

### See Also

[Using Initial Mesh Options](#)

[FEM Meshing Options](#)

## Specifying Advanced Tab Settings

Click the **Advanced** tab to display the advanced options:

The screenshot shows the 'Advanced' tab selected in a software interface. The 'Target Mesh Growth' is set to 25. There is a checkbox for 'Use Initial Minimal Mesh Size' which is checked, with a value of 0.3 mm and a button labeled 'Automatically determine'. Below this is a checkbox for 'Merge Objects with Same Material' which is unchecked. Further down is a checkbox for 'After convergence, solve on the finest mesh' which is checked. At the bottom, there is a section titled 'Automatic conductor mesh settings:' containing two checked options: 'Edge meshing' with a value of 0.2 and 'Vertex meshing' with a value of 0.2\*0.3, both followed by the text 'x estimated conductor width'.

You can specify the following advanced options:

Option	Description
Target Mesh Growth	

Option	Description
	Specifies the percentage by which each adaptive mesh grows relative to the previous mesh.
Use Initial Mesh Size	Enables you to set a minimum length of edges in the initial mesh. During mesh refinement, the individual length of edges in the mesh can be shortened where deemed necessary by the adaptive refinement process. If you click <b>Automatically Determined</b> , a value is automatically calculated based on the requested frequencies and geometric size. Using this option, you can restrict the meshing process to a more limited set of possible solutions. It is advised to only use this options when the initial mesh is unreasonably dense in some areas due to some combination of geometric features triggering that behavior.
Merge Objects with Same Material	This option controls if two objects that have the same material properties may be merged prior to meshing. This option can be used to reduce the number of tetrahedrons generated. Object merging is applied only when the objects also have the same mesh priority.
After convergence, solve on the finest mesh	Performs simulation on the finest mesh after convergence.
Automatic conductor mesh setting Edge meshing	This option is used when automatic meshing is applied. It determines the length of the edge mesh in relation to the estimated width of the conductor.
Automatic conductor mesh setting: Vertex meshing	This option is used when automatic meshing is applied. It determines the length of the vertex mesh in relation to the estimated width of the conductor.

**NOTE**

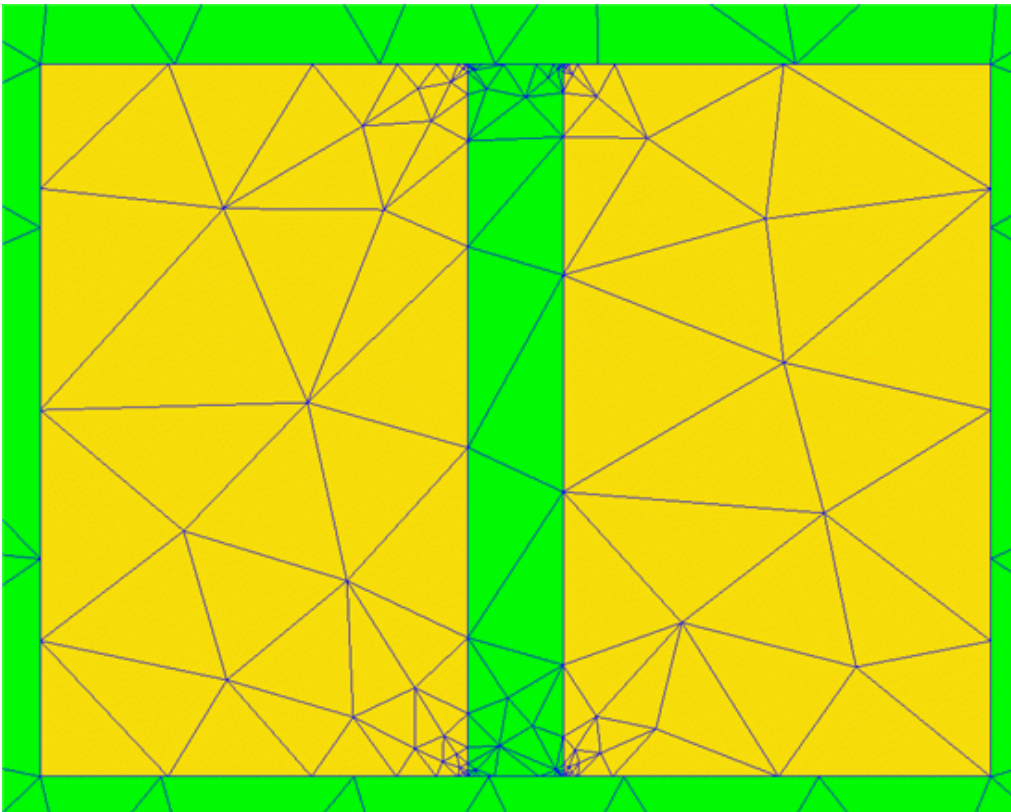
Advanced meshing options should be changed only in case of meshing problems. For more information about advanced meshing options, see [FEM Meshing Options](#)

## Vertex Mesh

Vertex meshing reduces the mesh length around vertices. When combined with edge meshing the shortest length of both is chosen in any given region. The region where the value is applied by the vertex mesh length setting is roughly twice its length in each direction: the actual region is still being optimized. The region affected by it depends on local geometry.

The following figure displays the result of a vertex mesh setting:

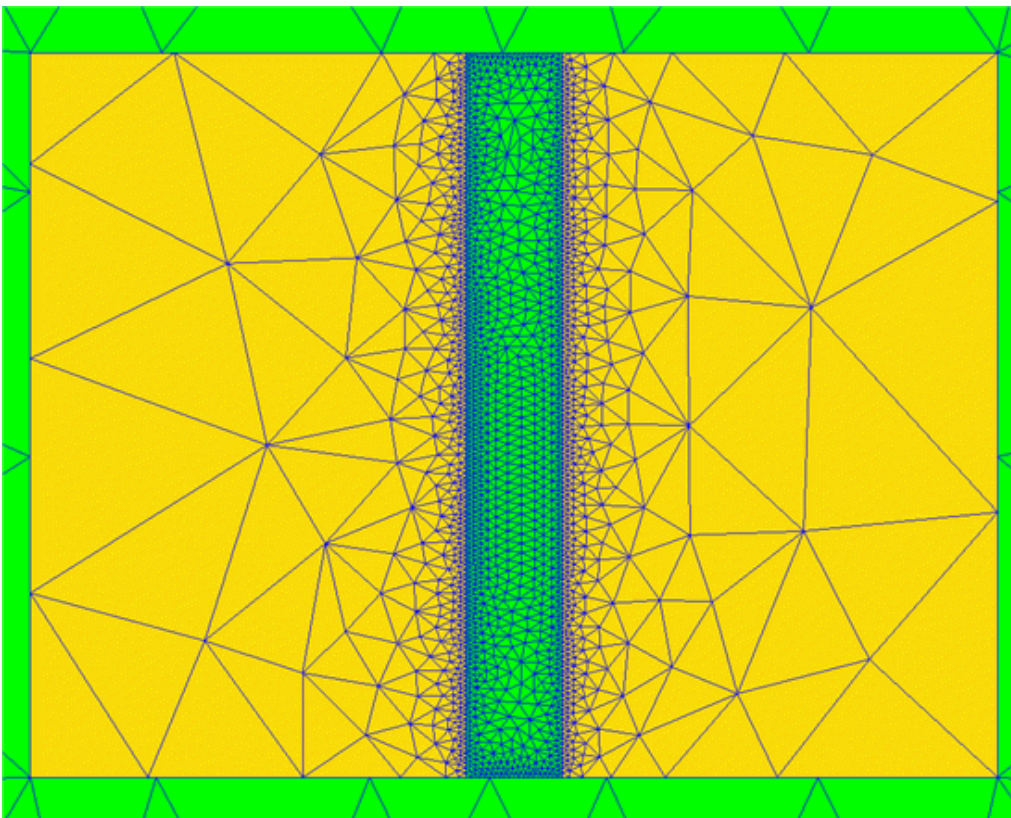




## Edge Mesh

Edge meshing reduces the mesh length around the edges. A fan-out effect is created to maintain the tetrahedron quality. The edges do not need to be straight lines, any shape is allowed.

The following figure displays the result of an edge mesh setting:




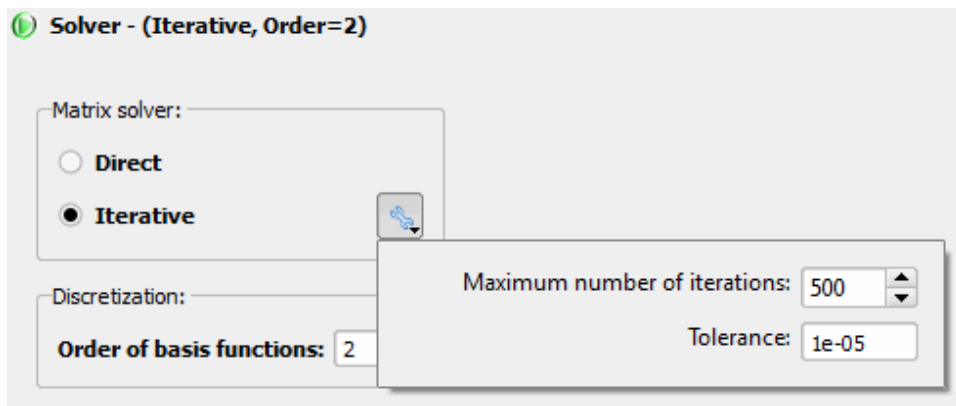


## Selecting a Solver in FEM

Using the Setup FEM Simulation window, you can specify the solver type settings and order of basis functions. You can select the following types of solver for your FEM simulation:

- **Direct:** When selecting **Direct**, the FEM Simulator will use a multithreaded sparse direct solver. The memory requirements and computing time of this solver typically scale as  $N^{(3/2)}$  to  $N^2$  respectively, where  $N$  is the matrix size. This solver does not suffer from potential convergence problems, and is guaranteed to yield a solution if sufficient memory is available.
- **Iterative:** When selecting **Iterative**, FEM Simulator will use the iterative matrix solver. The memory requirements and computing time of this solver scale as  $N$  to  $N^{(3/2)}$  respectively where  $N$  is the the matrix size. In other words, the computing resources of the iterative solver are typically one order of magnitude lower then the computing resources of the direct solver. Especially, the iterative solver requires significantly less memory then the direct solver. However, contrary to the direct solver, the iterative solver is not guaranteed to converge.

To increase the speed of FEM simulations, you can set advanced options for the Iterative solver. Click  to display the advanced options:



Contrary to the direct solver, the accuracy of the iterative solver can be chosen, by using the **Tolerance** option. If this option is decreased, the results will be more accurate. However, the number of iterations will increase, resulting in an increased solve time. The maximum number of iterations can be set by the **Maximal number of iterations** option.

## Order of Basis Functions Setting

The **Order of basis functions** is used to approximate the electric fields in all tetrahedras. You can set the following values for the *Order of basis functions* option:

- **2:** The second order basis functions will approximate the electric fields with a quadratic variation in each tetrahedron, and the magnetic fields will be approximated with a linear variation in each tetrahedron. This gives the best accuracy.
- **1:** The first order basis function will approximate the electric fields with a linear variation in each tetrahedron, and the magnetic fields will be approximated as being constant in each tetrahedron. This requires less memory than the second order basis functions given the same mesh, but typically requires a larger mesh to get the same level off accuracy. This setting is useful for electrically small structures.

It is recommended to use the second order basis function, which represents the fields at the edges of tetrahedrons. The advantage is that, for the 3D field distribution, less tetrahedrons are needed (they can be bigger). The first order basis function results in much less unknowns per tetrahedron compared to the second order. However, to achieve convergence, you need more and smaller tetrahedrons. In general, second order results in the smallest problem size for a given accuracy. In addition, when using the second order basis functions, you can use the iterative matrix solver option that typically requires less memory compared to the direct solver.

# Specifying FEM 2D Port Simulation Setup

Before running an FEM 2D Port simulation, you need to set the simulation options that are specific to the FEM 2D Port simulator. You can specify the simulation options by using the Setup FEM 2D Port Simulation window. This section provides information about how to run an FEM 2D Port simulation.

FEM 2D Port simulation enables you to:

- Determine the number of modes that are propagating on a port.
- Determine the relative field strength, impedance, and propagation constant for a mode on a selected port. If you do not have all propagating modes included for a simulation, errors are generated during simulation.

**NOTE**

If you want the simulation results to return relative field strength, impedance, and propagation constant, you must apply impedance lines to a port before running the simulation.


**See Also**

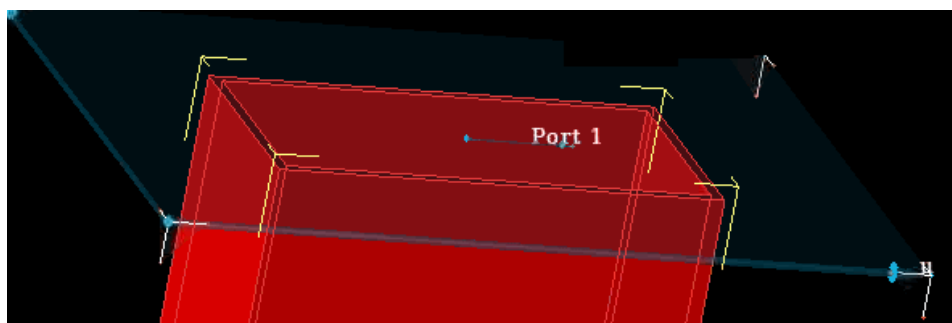
[FEM 2D Port Solver Overview](#)

## Adding Waveguide Ports

Waveguide ports enable energy to flow into and out of a structure, which is necessary as part of the simulation process. By default, a structure is assumed to be completely encased in a conductive shield with no energy propagating through it. You can apply ports to a structure to indicate the area where energy enters and exits the structure.

To add a waveguide port:

1. Right-click **Circuit Components/Ports** and choose **New Waveguide Port**. The **EMPro Waveguide Ports Editor** dialog box is displayed.
2. Click  to specify a face in the geometry.
3. Place the pointer on the top of the structure, as shown in the following figure:



All tabs in the **EMPro Waveguide Ports Editor** dialog box are enabled after you have specified a face in the geometry.

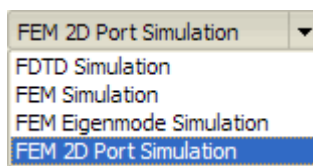
4. Click the **EditCrossSectionPage** tab.


5. By default, the **Auto-extend to simulation domain boundaries** option is selected. To customize boundary extensions, remove selection from this option and specify the lower and upper boundary extensions.
6. Click the **Properties** tab.
7. Type a waveguide port name in the **Name** text box.
8. Select **1W Modal Power Feed** from the **Waveguide Port Definition** drop-down list.
9. Select **Power/Current** from the **Impedance Definition** drop-down list.
10. Select a value from the **Number of Modes** drop-down list.
11. Click the **Impedance Lines** tab.
12. Specify value for the two endpoints.
13. Select a value from the **Mode Number** drop-down list.
14. Click **OK**.

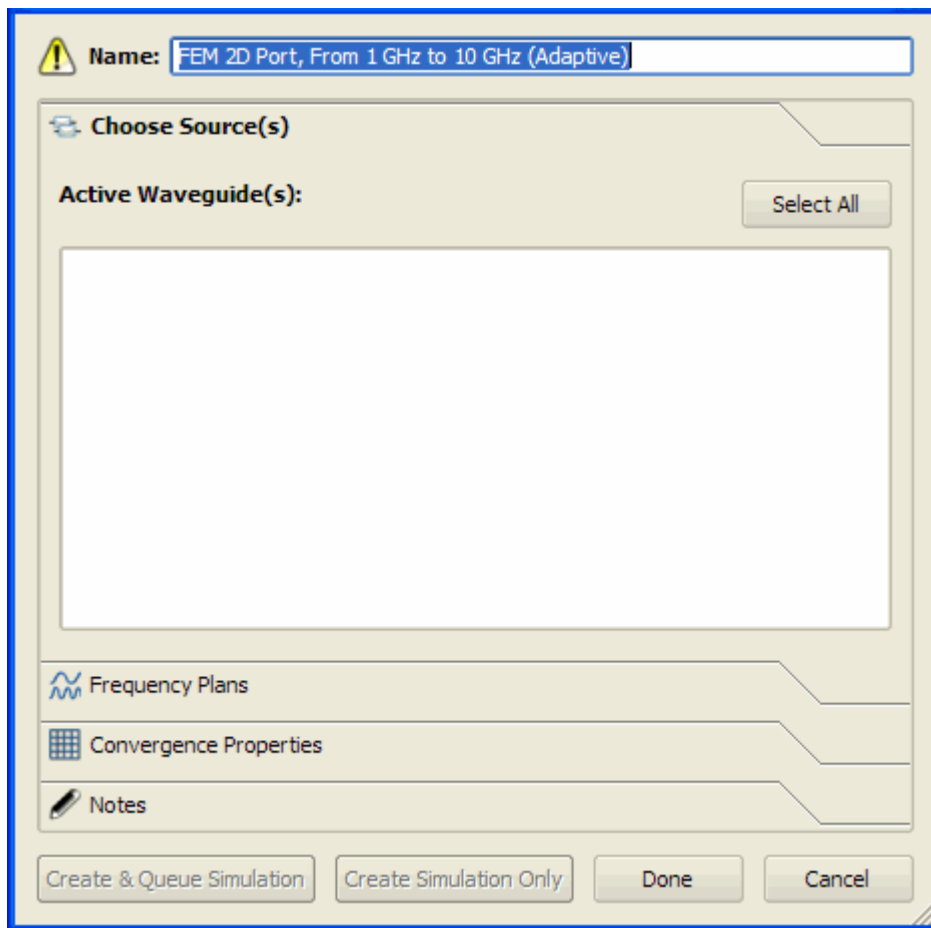
## Selecting FEM 2D Port Simulation

To select a FEM 2D Port simulation setup:

1. Choose **FEM 2D Port Simulation** from the drop-down list available on the **Simulation** toolbar, as shown in the following figure:



2. Click **Setup** (  ) to create and edit the setup of a new simulation. The Setup FEM 2D Port Simulation window is displayed, as shown in the following figure:

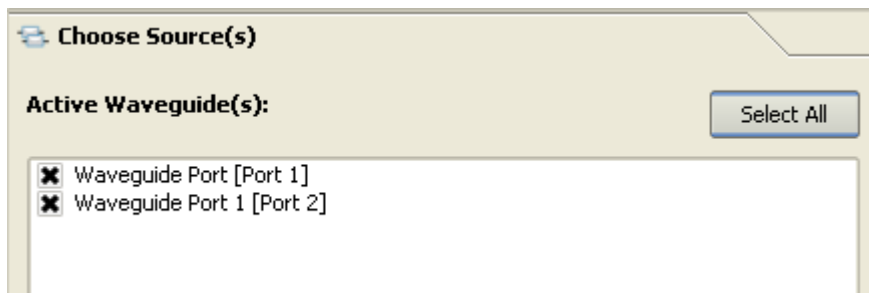


3. Type a name for your FEM 2D Port simulation in the **Name** text box. You can also use the default name that is specified in the Setup FEM 2D Port Simulation window. The invalid symbol on the window indicates that the setup for the FEM 2D Port Simulation is not valid. The set up might be invalid because of the following issues:
- No waveguide ports have been defined for this project.
  - No waveguide ports have been selected as active.
  - No frequency plan has been defined.
  - A material or boundary condition has been defined that is not supported for FEM 2D Port simulations.

## Selecting Sources

To choose waveguide ports:

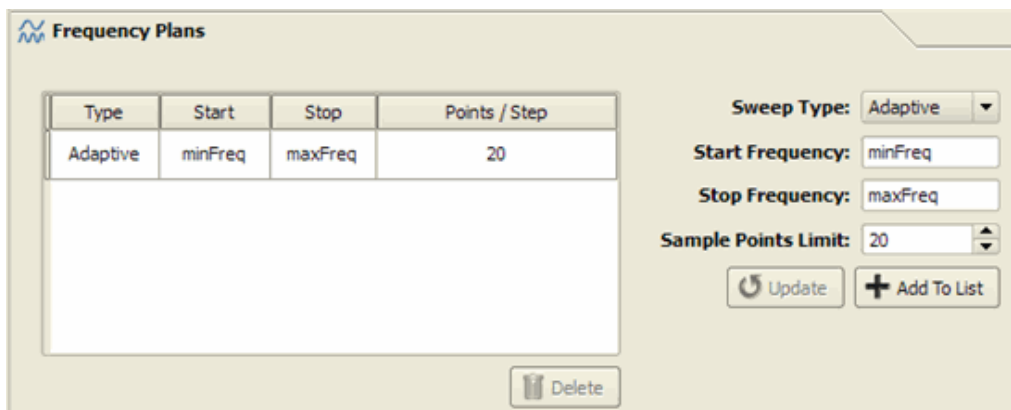
1. Open the Setup FEM 2D Port Simulation window. The Choose Source(s) screen is displayed, by default.
2. Select the waveguide ports for simulation.
3. You can also click **Select All** to choose all the waveguide ports listed in the **Active Waveguides** list. The following figure displays the Choose Source(s) screen:

**NOTE**

Only waveguide ports are listed in the Setup FEM 2D Port Simulation window.

## Specifying the Frequency Plans

Click **Frequency Plans** in the Setup FEM 2D Port Simulation window. This displays the options for setting your frequency plans, as shown in the following figure:



You can specify the following frequency settings for your FEM 2D Port simulation:

Option	Description
Sweep Type	Specify the type of sweep used for simulating ports. You can choose either Linear, Logarithmic, or Single option.
Start Frequency	Specify the start frequency value.
Stop Frequency	Specify the stop frequency value.
Sample Points Limit	Specify number of points.

Note that FEM Simulations and FEM 2D Port Simulations share the same frequency plan specification. Changes to the frequency plan made in this dialog will carry over to FEM Simulations for the full 3D structure.

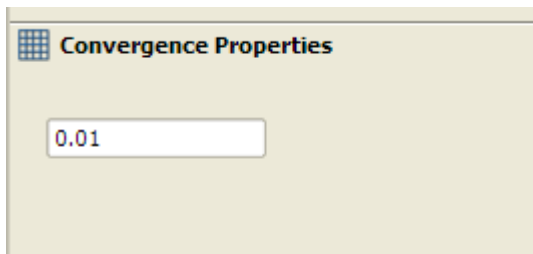
## Defining a Frequency Plan

1. In the Setup FEM 2D Port Simulation window, click **Frequency Plans**.

2. Select a value from the **Sweep Type** drop-down list.
3. Type a value in the **Start Frequency** text box.
4. Type a value in the **Stop Frequency** text box.
5. Select a value from **Sample Points Limit**.
6. Click **Add to List** to add a frequency plan. You can also modify the frequency plan by clicking **Update**. To delete a frequency plan, click **Delete**.

## Specifying Convergence Properties

Click **Convergence Properties** and specify a target value for the convergence. The target convergence value represents a relative error in the computed characteristic impedance. The FEM 2D Simulation will adaptively refine a 2D mesh at the highest frequency until two consecutive simulations produce characteristic impedances that differ by less than the requested accuracy. A target relative error of **0.01** would represent an absolute error of 0.5 Ohms for a 50 Ohm transmission line.



## Adding Notes

---

If you want to add any notes or observation with your simulation, you can specify it in the Notes text box.

1. In the Setup FEM 2D Port Simulation window, click the **Notes** tab.
2. Type a note that you want to be saved with the simulation.

## Completing the Process of Specifying Setup

---

After you have specified the frequency plans and mesh refinement options, click **Done** in the Setup FEM 2D Port Simulation window to apply the current settings. You can also click **Create Simulation Only** to accept the settings or **Create and Queue Simulation** to run and place the simulation in queue.

## Running FEM 2D Port Simulation

---

After completing the FEM 2D Port simulation setup, you can run calculations on the geometry. You can create, queue, and run simulations using the Simulations workspace. To run a FEM 2D Port simulation, click **Create and Queue** in the Simulations workspace.

## Running FEM 2D Port Simulation from the EMPro Waveguide Ports Editor Dialog Box

You can also launch an FEM 2D Port simulation for a single waveguide port from the **EMPro Waveguide Ports Editor** dialog box by clicking the **Simulate** button. Clicking the Simulate button launches an FEM 2D Port Simulation with the following settings:

- A default simulation name will be constructed as "FEM 2D Port: " followed by the name of the waveguide port being edited.
- Only the waveguide port associated with the open dialog box will be simulated.
- The frequency plan specified in the **Setup FEM 2D Port Simulation** window is used.
- The convergence properties specified in the **Setup FEM 2D Port Simulation** window is used.
- No notes will be generated.

E-field plots for all modes are automatically displayed when the FEM 2D Port simulation completes. H-fields, characteristic impedances, and propagation constants for the modes of this waveguide port can also be displayed manually.



# Specifying Eigenmode Simulation Setup

## Eigenmode Overview

You can use an Eigenmode solver to generate resonance properties (eigenmodes) of a closed structure without enforcing excitations. In the results, you can view the Eigen frequencies, Q value, Eigen field, and surface currents at each Eigen mode. This new solver quickly finds the resonant frequencies for devices such as cavity filters, which is a common high-frequency component used in wireless communication systems. Filter designers can also visualize the resulting electromagnetic fields at each resonant frequency and make adjustments to the cavity structure to optimize filter performance.

The Eigenmode solver is based on FEM technology and you need an FEM Simulator Element license. The existing EMPro FEM solver, regular mode solver, is driven by excitations and generates S parameters and/or radiation fields. The Eigenmode solver process is similar to a typical FEM or FDTD flow in 3D EM simulations.

### See Also

#### Eigenmode Overview

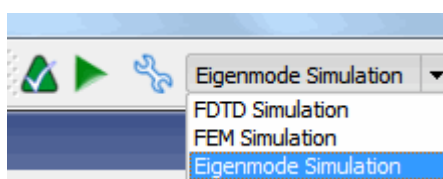
Before running an Eigenmode simulation, you need to specify the simulation options that are specific to the Eigenmode simulator. To run an Eigenmode simulation, perform the following tasks:


- Select the Eigenmode Solver.
- Specify Frequency Plans.
- Specify Mesh Refinement.
- Add Notes.
- Complete the Process of Specifying Setup.
- Run the Eigenmode Simulation

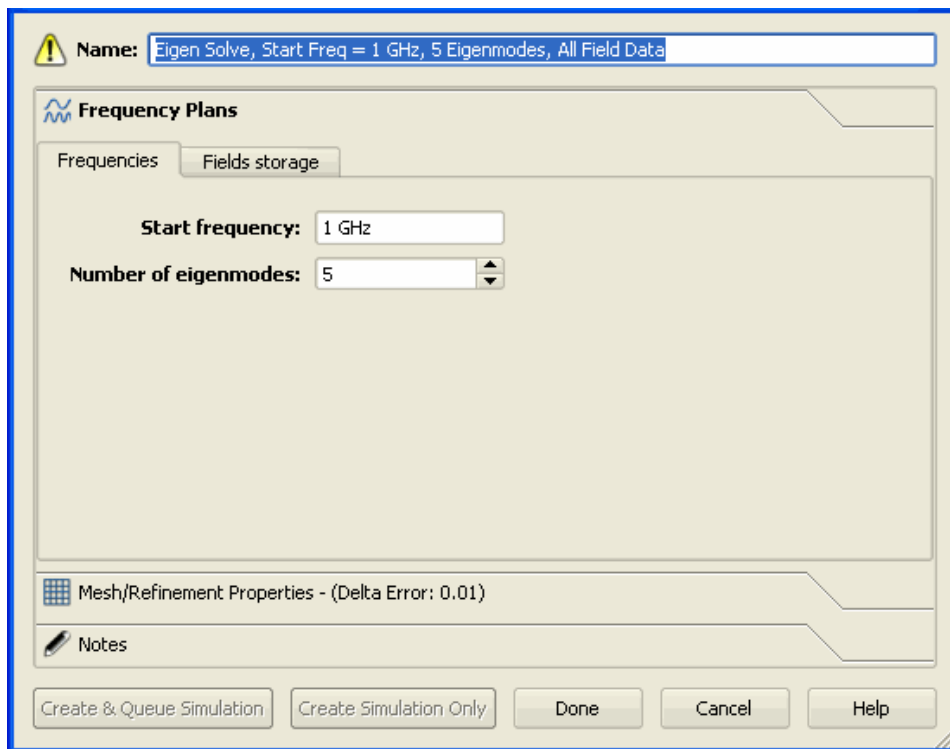
## Selecting the Eigenmode Solver

You can specify the simulation options that are specific to the Eigenmode simulator in the Setup Eigenmode Simulation window. To open the Setup Eigenmode Simulation window:

1. Select **Eigenmode** from the drop-down list available in the **Simulation** toolbar, as shown in the following toolbar:



2. Click **Setup** (  ) to create and edit the setup of a new simulation. The **Setup Eigenmode Simulation** window is displayed, as shown in the following figure:



3. Type a name for your Eigenmode simulation name in the **Name** field. You can also use the default name that is specified in the Setup Eigenmode Simulation window. The invalid symbol on the window refers that you have not entered any frequency plan or boundary conditions are not correct for Eigenmode simulations.

## Specifying Frequency Plans

You can define the following start frequency settings for your Eigenmode simulation:

Option	Description
Start Frequency	Specify the start frequency value.
Number of eigenmodes	Specify the number of Eigenmode in simulation

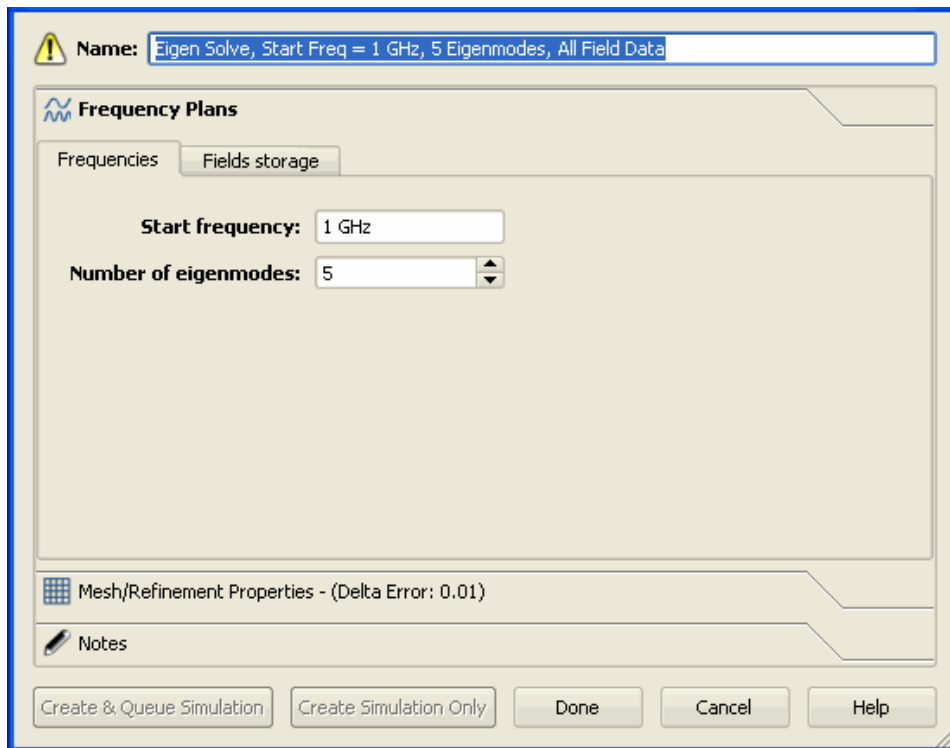
In the **Fields storage** tab, you can specify the options listed in the following table:

Option	Description
All eigen frequencies	Field data is stored for all eigen frequencies used in calculation.
No frequencies	Field data is not stored for any of the frequencies used in calculation.

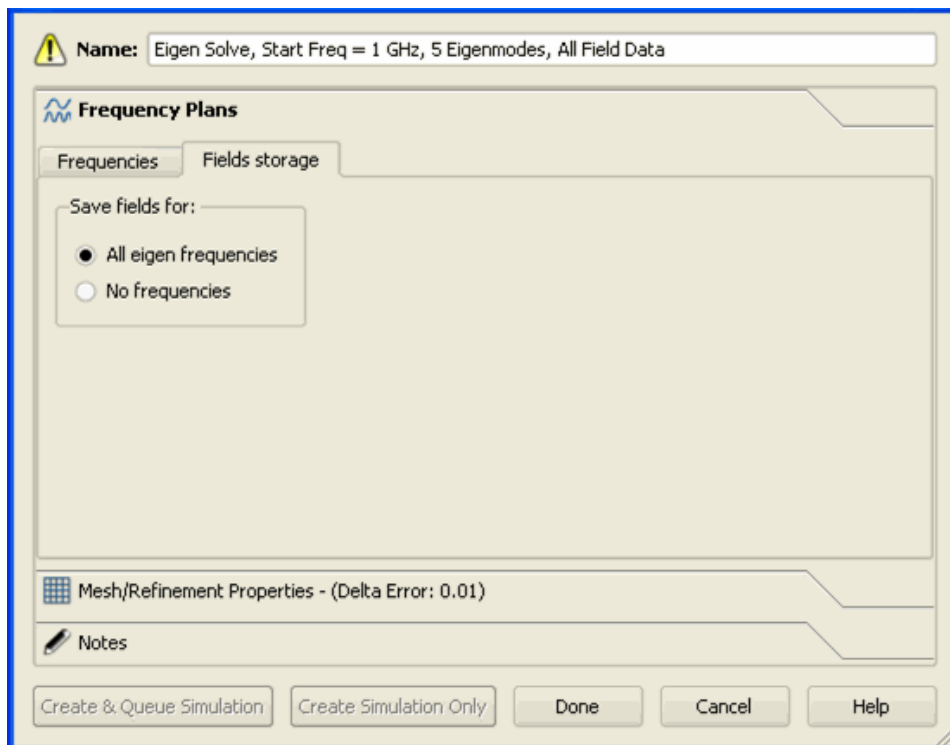
## Setting Frequencies

Perform the following steps to set frequency:

1. In the Setup Eigenmode Simulation window, open the **Frequency Plans** screen.
2. Specify a value in the **Start Frequency** text box.



3. Specify a value in **Number of Eigenmodes**.
4. Click the **Fields storage** tab. The following figure displays the field storage options:



5. Select one of the following options in **Save fields for**:
  - All eigen frequencies
  - No frequencies

## Specifying Mesh Refinement

An Eigenmode simulation mesh is a sub-division of the entire 3D problem domain into a set of tetrahedra (or cells). This pattern of cells is based on the geometry of a cavity and optionally, user-defined parameters, therefore, each cavity will have a unique mesh calculated for it. The mesh is then applied to the cavity to compute the electric within each cell and identify any coupling effects in the circuit during simulation.

The Eigenmode simulator implements an adaptive mesh algorithm, where an initial mesh is generated and the electric fields (and eigen frequencies) are computed on that initial mesh for a single frequency. An error estimate is generated for each tetrahedron. The tetrahedra with the largest estimated error are refined to create a new mesh on which the electric fields eigen frequencies are computed. The eigen frequencies from consecutive meshes are compared. If the eigen frequencies do not change significantly, then electric fields are computed for all the requested frequencies. However, if the frequencies do change significantly, then new error estimates are computed, a new mesh is generated and new electric fields and frequencies are computed.

Click **Mesh/Refinement Properties- (Delta Error: 0.01)** in the Setup Eigenmode Simulation window. This displays the **Mesh/Refinement Properties- (Delta Error: 0.01)** screen, which consists of three tabs: **Stop Criteria**, **Initial Mesh**, and **Advanced**.

### Stop Criterium Tab Settings

The following table describes the **Stop criterium** tab options:

Option	Description
Delta Error	Specifies the maximum allowed relative difference of corresponding eigenfrequencies between two successive refinement levels
Consecutive passes of delta error required	Specifies the required consecutive passes of delta error.
Minimum number of adaptive passes	Specifies the minimum number of refinement steps to be executed before the refinement process is allowed to stop.
Maximum number of adaptive passes	Specifies the maximum number of passes to be attempted. If the number of refinement passes entered is reached before the delta error criteria is met, the refinement process will end, based on this limit. The number of passes from all prior simulations is also displayed. Typically, a value between 10 and 20 is recommended.

### Initial mesh Tab Settings

Using the **Initial mesh** tab settings, you can control how the initial mesh is generated. The following table describes the **Initial mesh** tab options:

Option	Description
Use Initial Target mesh size	An initial denser mesh can be provided by checking this option. If used, no edge in the initial mesh will be longer than the given length. If <b>Automatically determine</b> is checked then EMPro automatically determines the initial target mesh size. The length is computed based on the frequency plan available when the request is made.
Use Initial minimum mesh size	This option can be used to set a minimum length of edges in the initial mesh. During mesh refinement the individual length of edges in the mesh can be shortened where deemed necessary by the adaptive refinement process. When <b>Automatically determined</b> is selected an estimate is made of what is a good value for this mesh setting based on the requested frequencies and geometric size. Using this options restricts the meshing process to a more limited set of possible solutions. It is advised to only use this options when the initial mesh is unreasonably dense in some areas due to some combination of geometric features triggering that behavior.
Conductor edge mesh length	Specifies a target mesh size of all edges that belong to the geometry of a conductor. Edges do not necessarily have to be straight lines.
Conductor vertex mesh length	Specifies a target mesh size for all vertices that belong to the geometry of a conductor.

For all lengths a parameter or formula is allowed. The formula is always evaluated at the time the simulation is being created.

## Advanced Tab Settings

The following table describes the **Advanced** tab settings:

Option	Description
Target Mesh Growth	Specifies the percentage by which each adaptive mesh grows relative to the previous mesh.
Use Initial Minimal Size	Specifies the minimum initial mesh size
Merge Objects with Same Material	Merges objects that are assigned with the same material

Automatic conductor mesh settings	<p>Specifies the following types of meshing:</p> <ul style="list-style-type: none"> <li>• Edge meshing</li> <li>• Vertex meshing</li> </ul>
-----------------------------------	---

## Guidelines

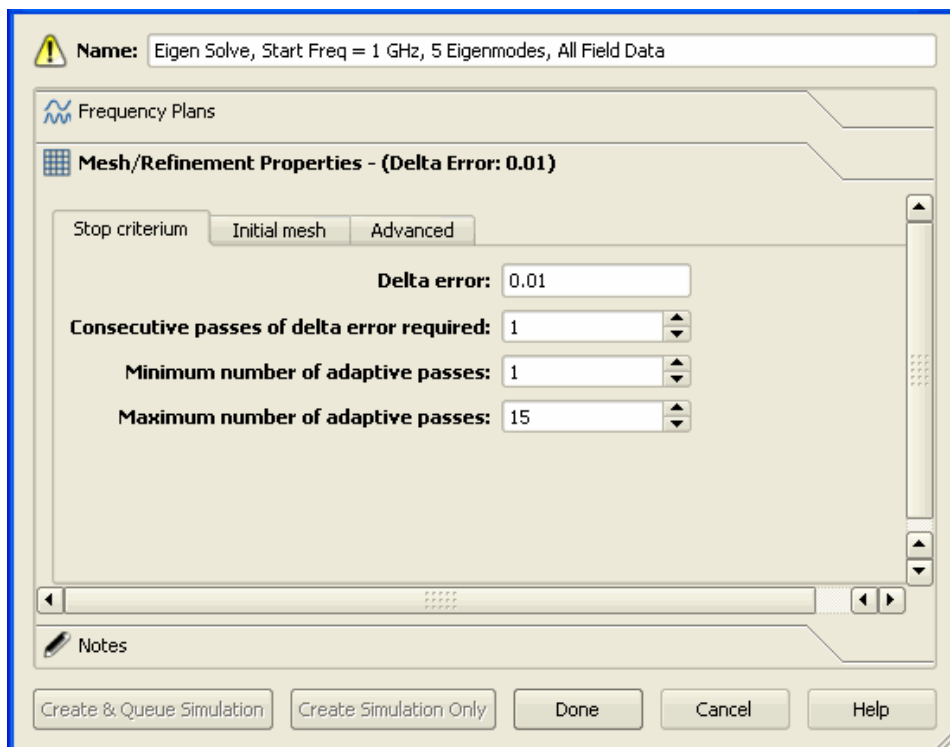
In the Eigenmode solver, it is important to note that:

- A **Delta error** value specifies the relative difference of the Eigen frequencies (delta frequency) between two consecutive passes.
- To generate an efficient mesh, enable **Use Initial Mesh Size** and **Use Initial Minimal Mesh Size** in the **Initial mesh** tab. The value in **Use Initial Mesh Size** should be between  $\lambda/3$  to  $\lambda/4$ , where  $\lambda$  is the free-space wavelength at lowFreqLimit.

## Setting Mesh Refinement Options

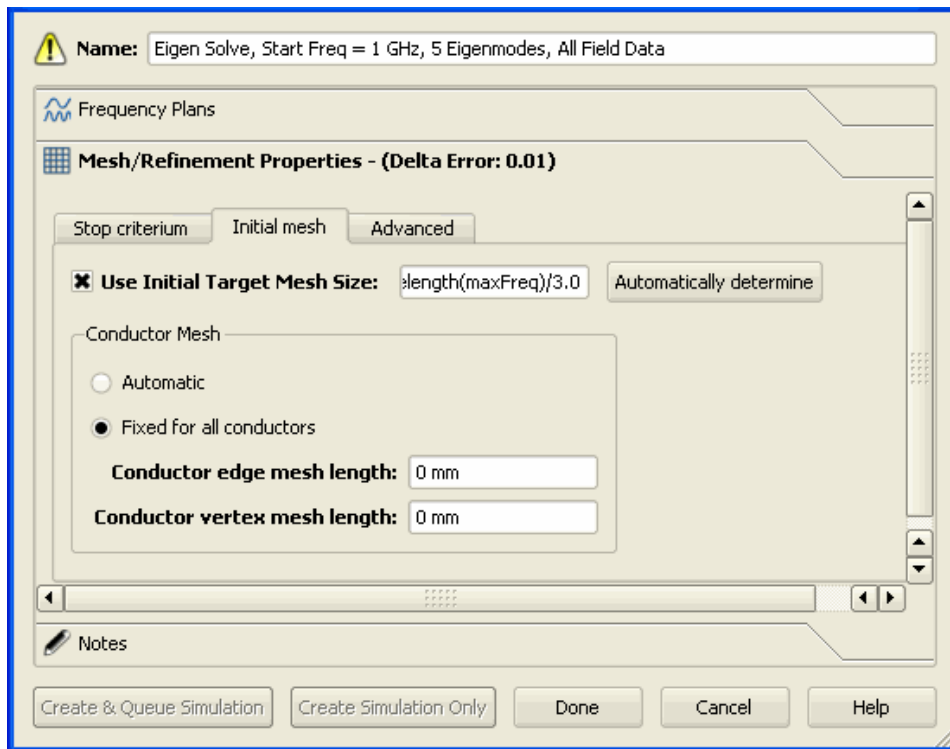
To specify the mesh refinement options:

1. Click **Mesh/Refinement Properties - (Delta Error:0.01)** in the Setup Eigenmode Simulation window.
2. The **Stop Criterium** tab is displayed by default, as shown in the following figure:

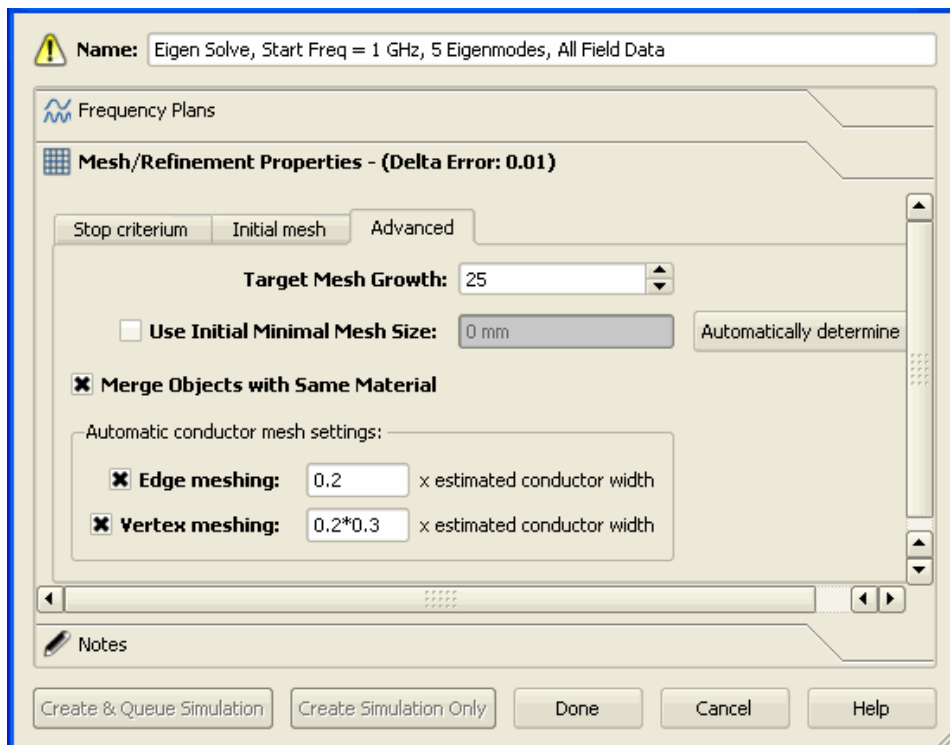


3. In the **Stop Criterium** tab:
  - Type a value in the **Delta Error** text field.
  - Specify a value in **Consecutive passes of delta error required**.
  - Specify a value in **Minimum number of adaptive passes**.

- Specify a value in **Maximum number of adaptive passes**.
4. Click the **Initial Mesh** tab. The initial mesh options are displayed, as shown in the following figure:



5. In the Initial Mesh tab:
- Select the **Use Initial Target Mesh Size** check box.
  - Type a value in the **Conductor edge mesh length** text field.
  - Type a value in the **Conductor vertex mesh length** text field.
6. Click the **Advanced** tab. The Advanced options are displayed, as shown in the following figure:



7. In the **Advanced** tab:

- Specify a value in **Target Mesh Growth**.
- Select the **Use Initial Minimal Size** check box and type a value. You can also click **Automatically determine** to update a value in Use Initial Minimal Size.
- Select the **Merge Objects with Same Material** check box.

8. In **Automatic conductor mesh settings**:

- Select **Edge meshing** and type a value.
- Select **Vertex meshing** and type a value.

## Adding Notes

If you want to add any notes or observation with your simulation, you can specify it in the Notes text box. Click **Notes** in the Setup Eigenmode Simulation window to display the **Notes** screen.

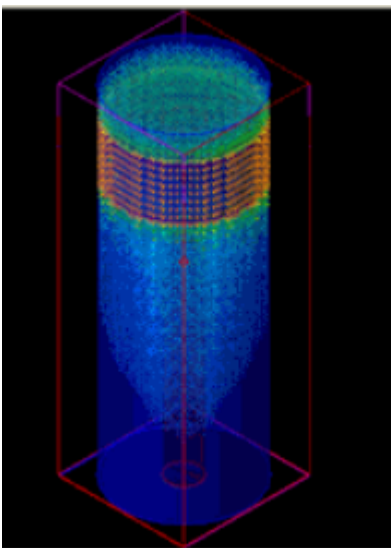
## Completing the Process of Specifying Setup

After you have specified the frequency plans and mesh refinement options, click **Done** in the Setup Eigenmode Simulation window to apply the current settings. You can also click **Create Simulation Only** to accept the settings or **Create and Queue Simulation** to create and queue the simulation.

## Running Eigenmode Simulation

After completing Eigenmode Simulation setup, you can run calculations on the geometry. The Simulations workspace window stores the project simulation(s). You can create, queue, and run simulations using the Simulations workspace.

To run an Eigenmode simulation, click **Create and Queue** in the **Simulations** workspace. The following figure displays the visualization results of an Eigenmode simulation:





# Creating a Mesh

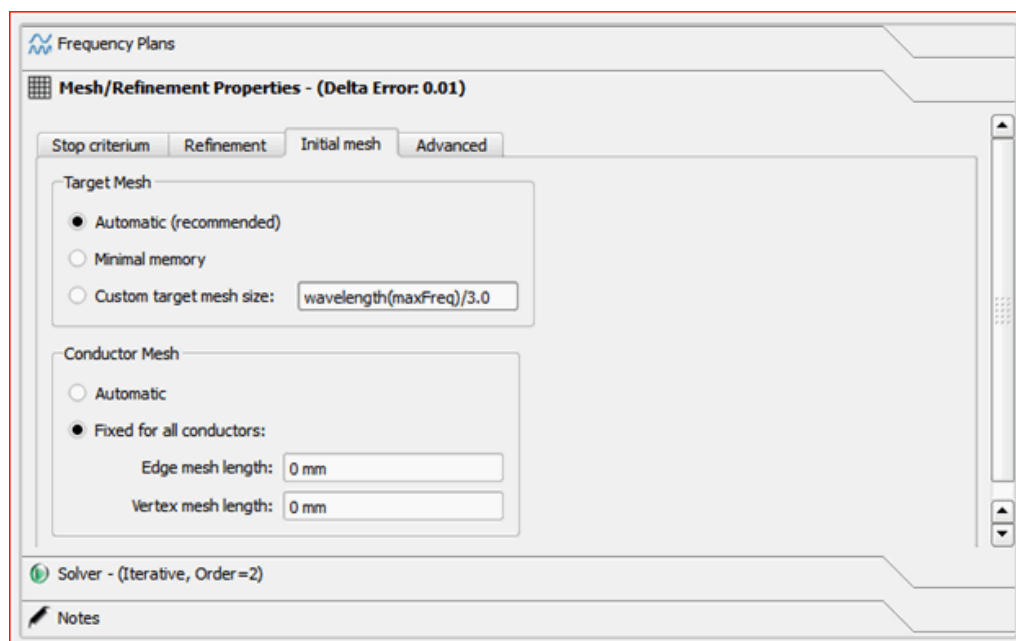
You can specify vertex mesh length, edge mesh length, and surface mesh length for your FEM simulations. In addition, you can automate the FEM broadband refinement process by selecting the required refinement strategy.

This section provides information about the following topics:

- [Initial Mesh Settings](#)
- [Performing FEM Broadband Refinement](#)
- [Viewing FEM Mesh](#)
- [Reusing Mesh and Frequency Points](#)
- [Troubleshooting Mesh Failures](#)

## Initial Mesh Settings

The initial mesh is the starting point for the mesh refinement process, part of the FEM simulation. The “Initial mesh” settings allow you to control the first mesh.



## Target Mesh Options

The Target Mesh specifies the size of the edges on tetrahedra. You can specify the following Target mesh options:

### Automatic

Automatic is the recommended setting. The target mesh length will be one third of the free space wavelength at the maximum frequency. It is a compromise between accuracy and simulation performance.

The target mesh length is used for the full simulation domain. No tetrahedron in the simulation domain will contain an edge which is longer than the target mesh length. If there is a large area of the simulation domain that does not contain any fields, then using a global target mesh length can generate too many tetrahedral and it is advised to revert to the “custom target mesh size” setting or use the “minimal memory”.

### **Minimal Memory**

When Minimal Memory is chosen, the initial mesh will generate a mesh with the least amount of tetrahedra as possible. A typical mesh is then determined by the geometric detail of the design, the wavelength does not play any role at all. The advantage of using this setting is that no part of the design will contain more tetrahedral than necessary. However, there might be insufficient tetrahedra to have an accurate simulation result. When using the “Minimal Memory” it is recommended to either:

- have a sufficient number of mesh refinement steps.
- or to further customize the mesh with additional dedicated settings.

### **Custom Target Mesh**

The custom target mesh works similar to the automatic target mesh except that as a user you have the liberty to specify the mesh size. This choice can be used when the automatic derived mesh length is not sufficient. For example: not the maximum frequency but another frequency has to be used, or the design contains little to no free space and a different wavelength is important. The specification of the mesh size accepts a formula or expression. For example, specifying the custom target mesh size to be the 1/3 of the wavelength in free space at 2 GHz is done by entering as expression: “wavelength(2 GHz)/3”.

## **Conductor Mesh Options**

“Conductor Mesh” specifies the mesh lengths on conductive objects in the mesh. To obtain an accurate simulation result, typically the mesh on conductive objects is finer than on its surrounding elements.

There are two ways to specify the mesh lengths to be applied to conductors:

### **Automatic**

If you select the Automatic conductor mesh option, during the simulation process, the DC connectivity of the design is extracted and all conductive objects connected to a port are considered. The actual width of the conductive object is determined at each pin of a port. The estimated width is reported at the start of the mesh generation process. That width is used to set the edge mesh length on the connected pieces of conductive objects. The length set for the edges is by default 1/5 of the width. This factor can be tweaked on the “Advanced” tab of the “Mesh/Refinement Properties”.

Automatic conductor mesh performs a best-effort to determine the widths at all conductors and will report when there was a failure. The effective port/pin location contributes to the success of the estimation. Pins located at the edges of a conductor help disambiguating the width. The automatic conductor mesh will typically back out when a pin is connected to a curved surface or curved edge.

### **Fixed for all Conductors**

All conductors receive the specified mesh length. In contrast with the automatic conductors is there no difference between a conductor attached through a conductive path to a pin and a so called floating conductor.

This option is typically used when the automatic conductor meshing is not able to extract the correct widths or when there are conductors that need to be finer meshed that are not DC connected, but are at higher frequencies: a typical example is a comb like filter.

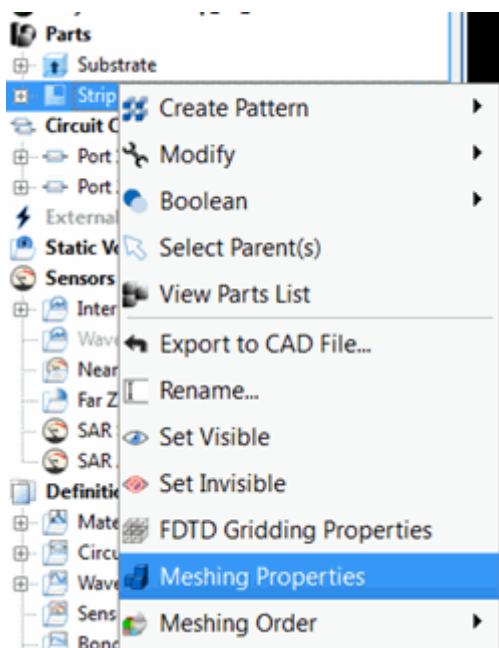
## Further Customizing Mesh Lengths

You can also set Edge, vertex and surface mesh length per object. Objects settings take higher priority but only when a smaller mesh length is specified. It is not possible to override an edge length per object with a length larger than the globally applied setting.

A value of “0” implies that the setting is ignored. Negative values will result in an invalid simulation setup. The values may be written as an expression using parameters if applicable.

EMPro provides advanced FEM mesh control settings to improve the accuracy of simulation results and ensure an efficient adaptive meshing. You can set FEM mesh options in the Setup FEM Simulation window or by using the Meshing Properties Editor. The Surface mesh length option is applicable to only specific objects.

To specify per-object settings, right-click the required object and select the “Meshing Properties” of an object. For more information, see [Initial Mesh Settings](#).



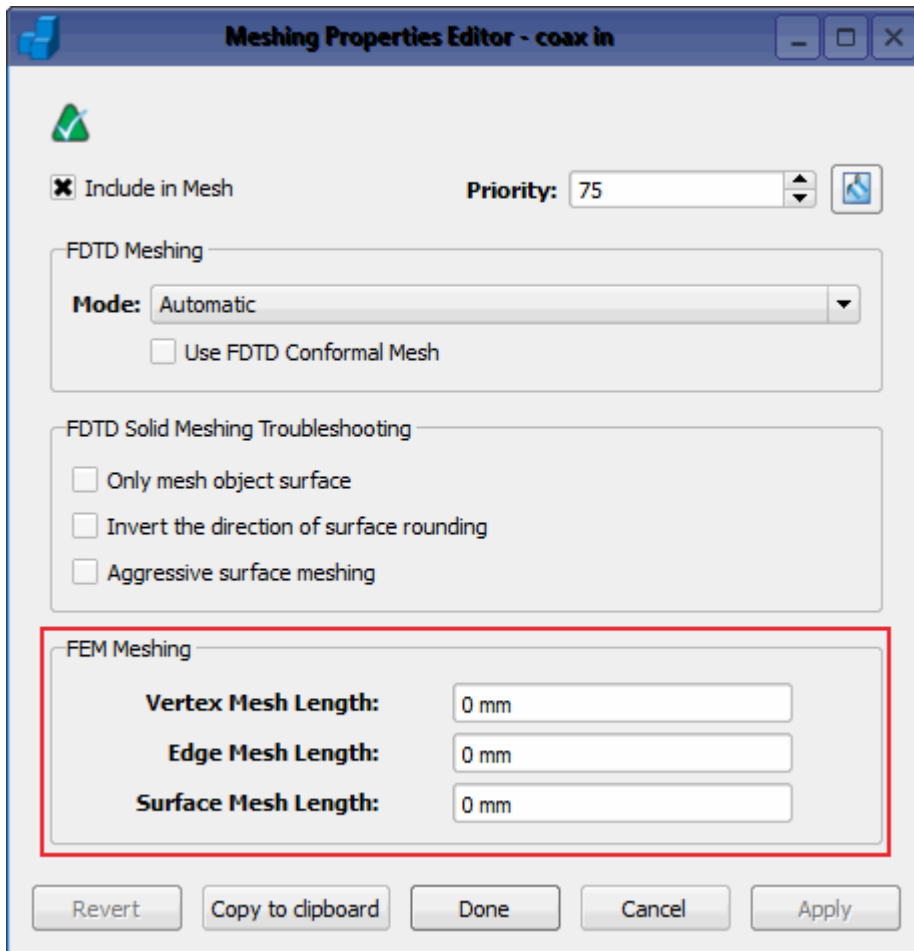
### NOTE

The advanced FEM mesh control settings provide the following benefits:

- Ensures an efficient adaptive meshing behavior: The additional mesh controls build a finer initial mesh so the adaptive meshing does not need to go through a large number of steps to arrive at the same level of mesh elements. Areas mainly around conductors are susceptible to under meshing during initial meshing. With the finer mesh controls, some of this knowledge can be injected by applying additional mesh settings to conductors but also at the individual object level.
- Improves the precision of the simulation results: The quality of the final mesh after adaptive meshing is higher. With the additional mesh settings known at the initial meshing stage this step can take all the requested constraints into account while during adaptive meshing quality properties of the mesh are balanced against mesh performance and electromagnetic constraints. The edge and vertex meshing in particular allow to better model singular effects at material interfaces.

## Meshing Properties Editor

You can select a mesh by using the Meshing Properties Editor. To open this editor, right-click an object and select **Meshing Properties**.



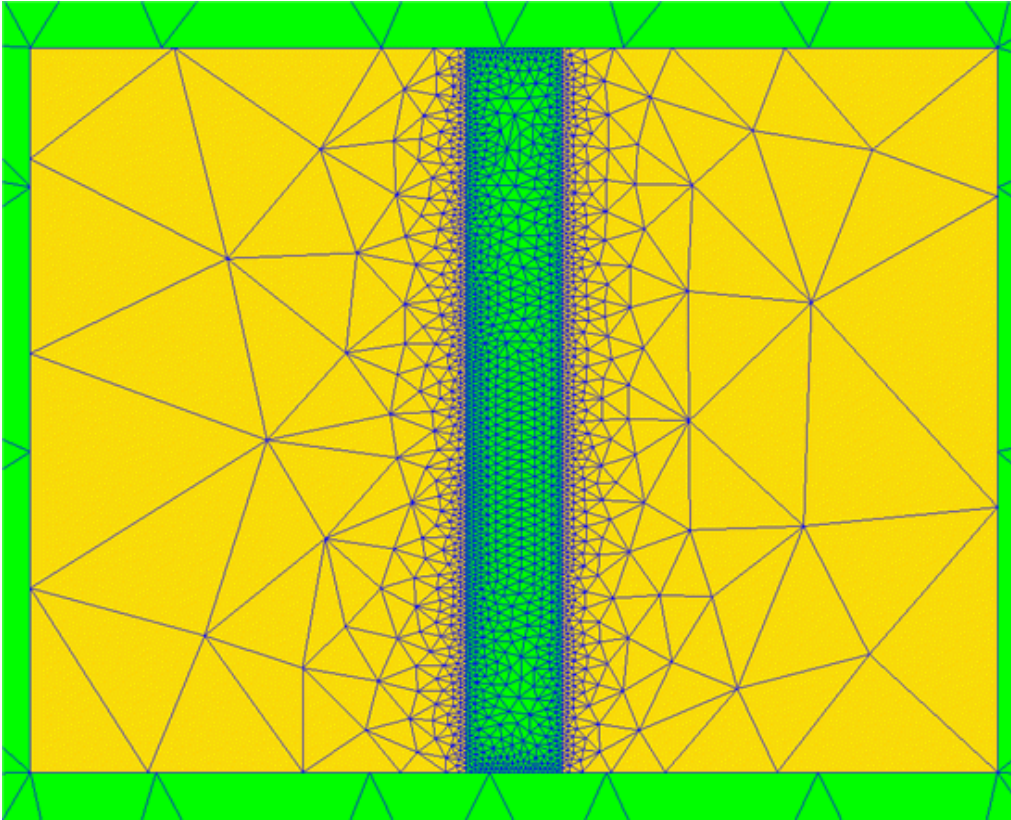
## Group Application of Mesh Settings

It is possible to edit the mesh settings of multiple objects simultaneously. The technique is similar to other parts of the EMPro GUI: paste the settings on objects. For copying mesh settings:

1. Open the Meshing Properties Editor by right-clicking an object and selecting **Meshing Properties**.
2. Click **Copy to Clipboard**.
3. After the copy operation is done, select a group of parts and request the context menu to perform the “Paste” action (or using the shortcut).

## Edge Meshing

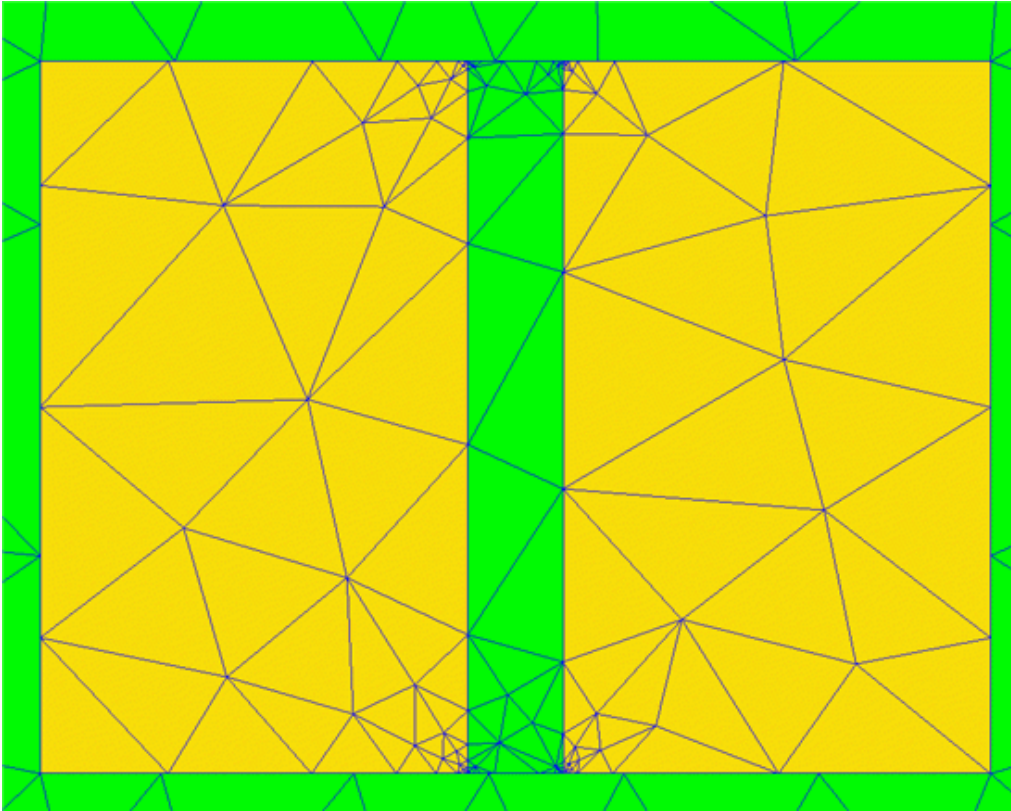
Edge meshing reduces the mesh length around the edges. In the following figure, the edge mesh length is specified as 0.1 mm and mesh is generated by applying an edge mesh to the strip line running on the substrate:



In this figure, around the edges of the strip, the length of mesh edge is reduced to 0.1 mm. A fan-out effect is created to maintain the tetrahedron quality. The edges do not need to be straight lines, any shape is allowed. A good value is one fifth of the width of a strip.

## Vertex Meshing

Vertex meshing reduces the mesh length around vertices. In the following figure, the vertex mesh length is set to 0.033 mm:



When combined with edge meshing, the shortest length of both meshes is chosen in any given region. The region where the value is applied by the vertex mesh length setting is roughly twice its length in each direction: the actual region is still being optimized. The region affected by it depends on local geometry.

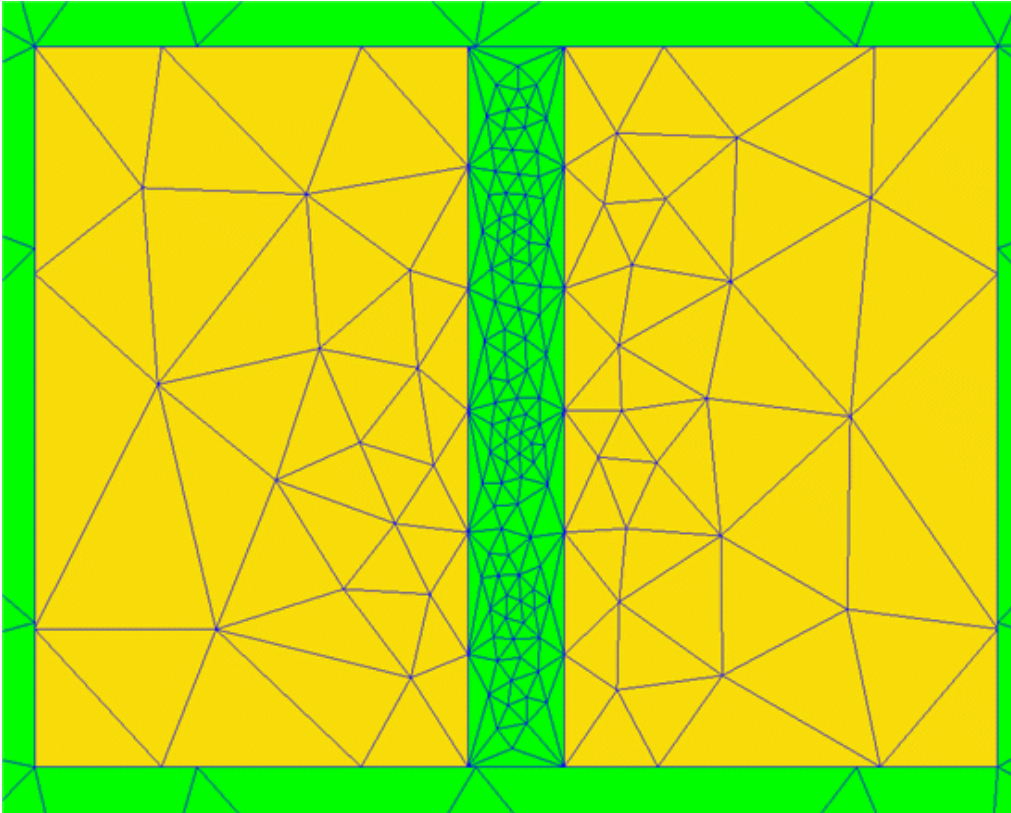
**NOTE**

Vertices in EMPro are zero-dimensional entities, like edges are one-dimensional. You can recognize vertices in straight line designs as the corners of shapes. During sketching vertices will advertise themselves as vertex0, vertex1, when hovering over them.

## Surface Meshing

Surface meshing reduces the mesh length on surfaces of objects. In the following figure, the surface mesh length is set to 0.5 mm, but the mesh length is not exactly 0.5 mm everywhere on the strip:

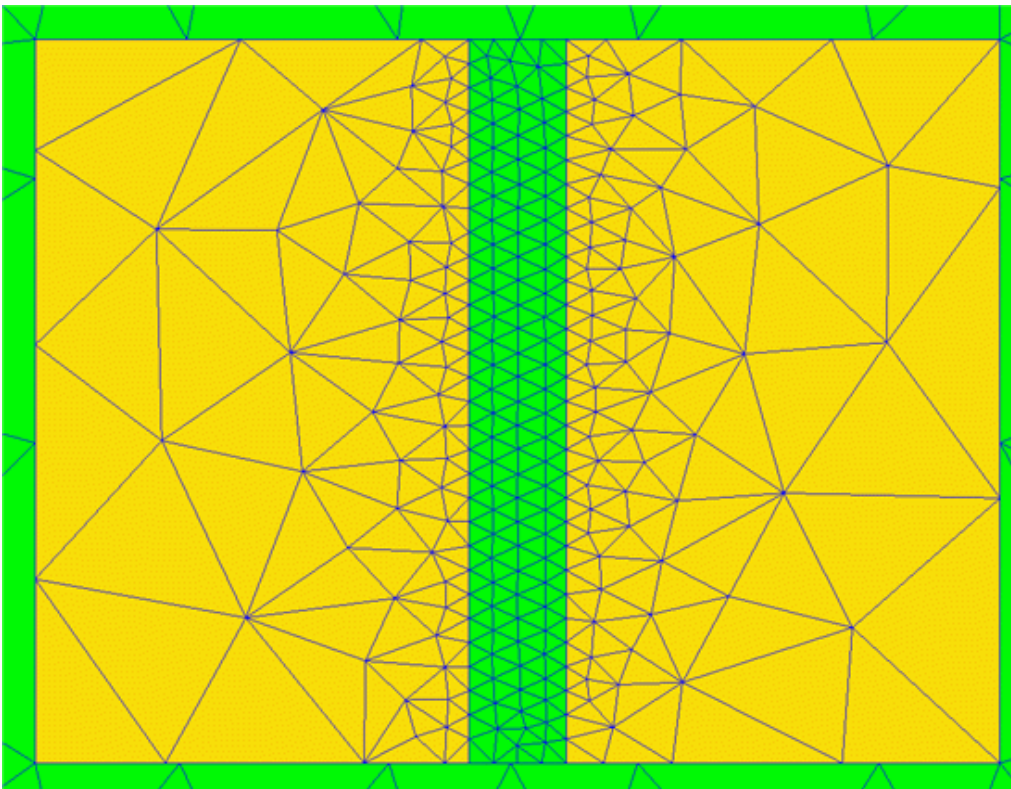




This effect is caused by the fact that the edge mesh length was not explicitly set. In that case, internally, the edge mesh length is reduced to five times the surface mesh length. This is necessary to maintain mesh quality and keep the performance of the meshing process under control. This option is useful when the affected region of the mesh control settings needs to be limited. The surface mesh length is not affecting other objects nearby the strip.

## Combined Edge and Surface Meshing

In the following figure, the edge and surface mesh length have both been set to 0.5 mm:



The main difference with the surface meshing applied in isolation is that the fan out of the meshing reaches into the substrate.

## General Settings and GUI

Edge and vertex mesh length can be set globally and are part of the simulation settings. Objects settings take higher priority, but only when they dictate a smaller mesh length. It is not possible to override an edge length per object with a length larger than the globally applied settings. A value of 0 means that the setting is ignored. Negative values will result in an invalid simulation setup. The values may be parameters if desirable. The per object settings are reachable through the context menu and are part of the meshing properties of an object.

### NOTE

You can edit the mesh settings of multiple objects simultaneously by copying the required settings on objects. In the Meshing Properties Editor, you can use **Copy to Clipboard** button. After the copy operation is done, it is possible to select a group of parts and then request the context menu and perform the “Paste” action (or using the shortcut).

### See Also

[Specifying FEM Simulation Setup](#)

## Using Initial Mesh Options

---

You can use the following mesh control options for controlling the initial mesh used in the adaptive refinement process:

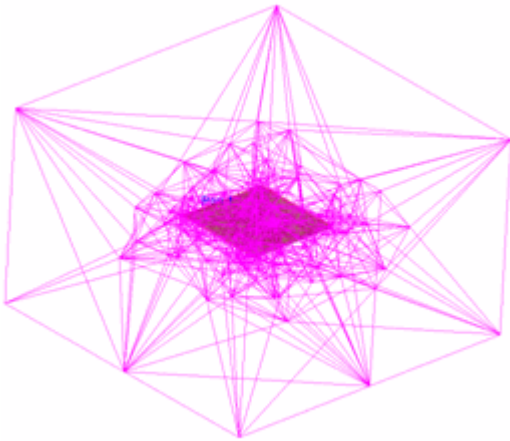
- Initial target mesh size
- Initial minimum mesh size

### Initial Target Mesh Size

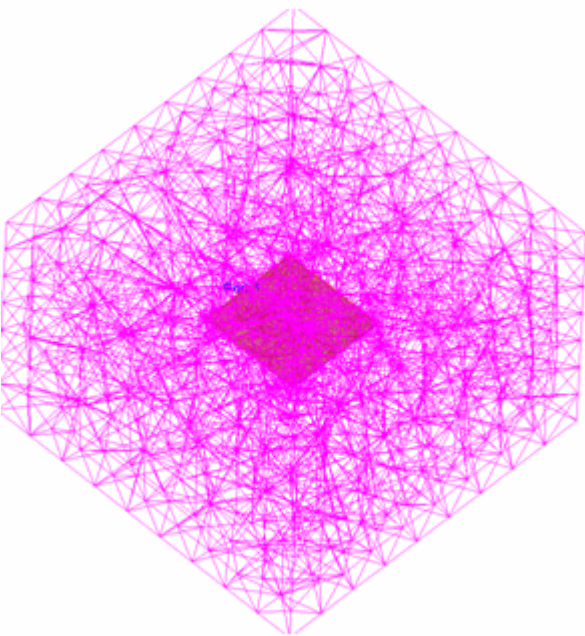
For electrically large structures, it is beneficial to fill the free space with a larger number of tetrahedra than based on geometric features alone. To accomplish this, the maximum size of edges in the initial mesh can be set to an appropriate value. Using the **Initial Mesh** tab of the *Mesh/Refinement Properties* window, specify a length in the **Use Initial Target Mesh Size** field. If you select the **Automatically Determine** option, EMPro provides a suggestion based on the given frequency plan. The automatically determined value is the wavelength in free space divided by three.

In this section, the Microstrip Dipole Antenna example is used. Open this example by selecting **Help > Examples > Microstrip Dipole Antenna**. The following figure displays the mesh generated over freespace when **Initial Target Mesh Size** is not used:





If the same volume is meshed by specifying the **Initial Target Mesh Size** option, the following mesh results are displayed:



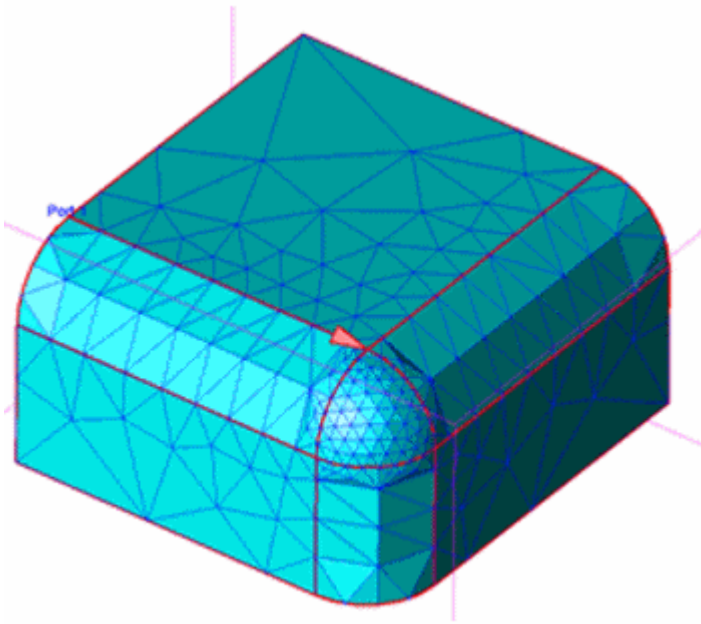
This leads to a faster convergence of iterative solver and in less number of passes in the Adaptive Frequency Sweep.

## Initial Minimum Mesh Size

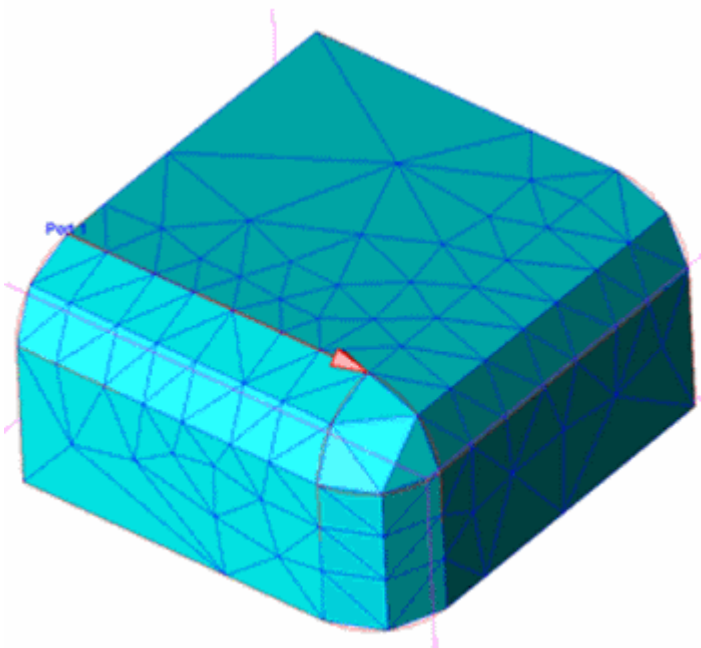
The Initial Minimum Mesh Size option controls the smallest length present in the initial mesh. It can be applied when there are geometric features present that are of less importance for the EM simulation. The result is a lower number of tetrahedra necessary to converge to the final solution. A typical use scenario for this feature is when a complex CAD model is imported or drawn where the geometric detail exceeds the detail required for the EM solution. The mesher is most effective in applying this constraint on curved surfaces.

Using the **Advanced** tab of **Mesh/Refinement Properties**, you can specify a length. If you select the **Automatically determine** option, the EMPro GUI will provide a suggestion based on the size of the geometry and value of the **Initial Target Mesh Size**. The ratio between the **Initial Target Mesh Size** and **Initial Minimum Mesh Size** cannot be lower than 10 to provide the mesher enough freedom to fill the solution space with tetrahedra.

The following figures illustrates the effect on a simple example when the setting minimum size is not set:



The following figures illustrates the effect on a simple example when the setting minimum size is set to 3 mm:



## FEM Meshing Options

EMPro provides advanced FEM mesh control settings to improve the accuracy of simulation results and ensure an efficient adaptive meshing. You can set FEM mesh options in the Setup FEM Simulation window or by using the Meshing Properties Editor. The following additional mesh controls are supported in FEM:

- Edge mesh length
- Vertex mesh length
- Surface mesh length

**NOTE**

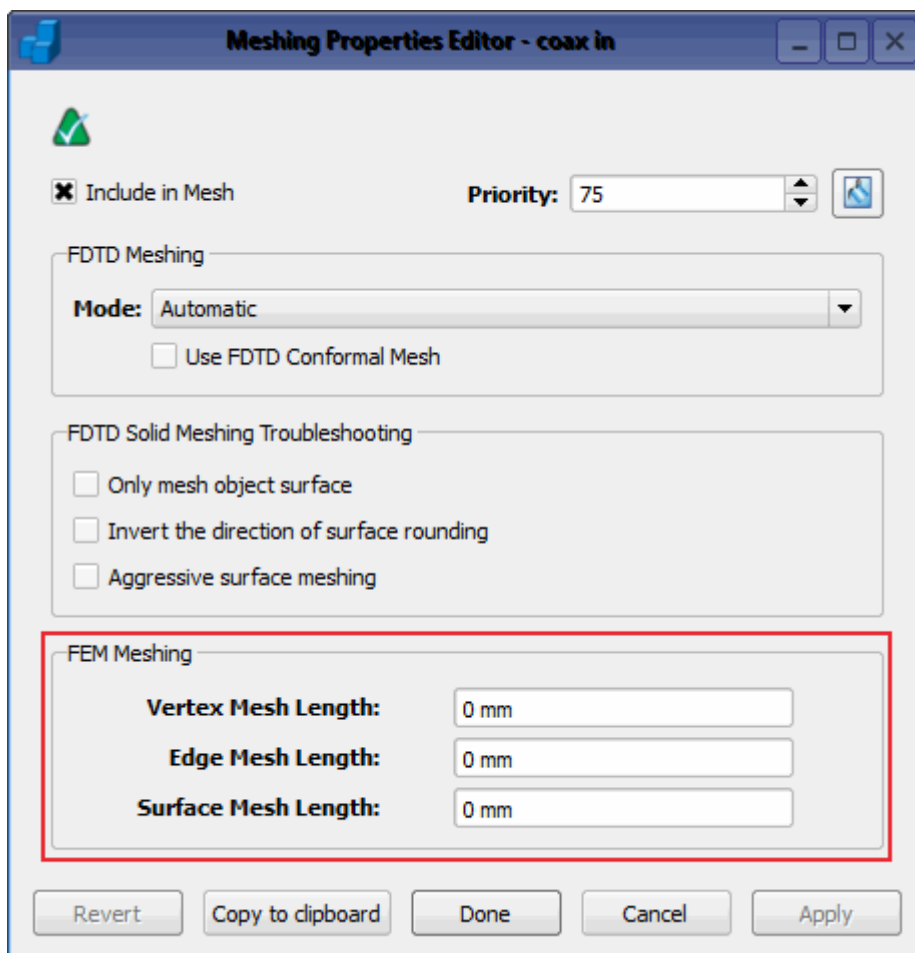
The Surface mesh length option is applicable to only specific objects.

The advanced FEM mesh control settings provide the following benefits:

- Ensures an efficient adaptive meshing behavior: The additional mesh controls build a finer initial mesh so the adaptive meshing does not need to go through a large number of steps to arrive at the same level of mesh elements. Areas mainly around conductors are susceptible to under meshing during initial meshing. With the finer mesh controls, some of this knowledge can be injected by applying additional mesh settings to conductors but also at the individual object level.
- Improves the precision of the simulation results: The quality of the final mesh after adaptive meshing is higher. With the additional mesh settings known at the initial meshing stage this step can take all the requested constraints into account while during adaptive meshing quality properties of the mesh are balanced against mesh performance and electromagnetic constraints. The edge and vertex meshing in particular allow to better model singular effects at material interfaces.

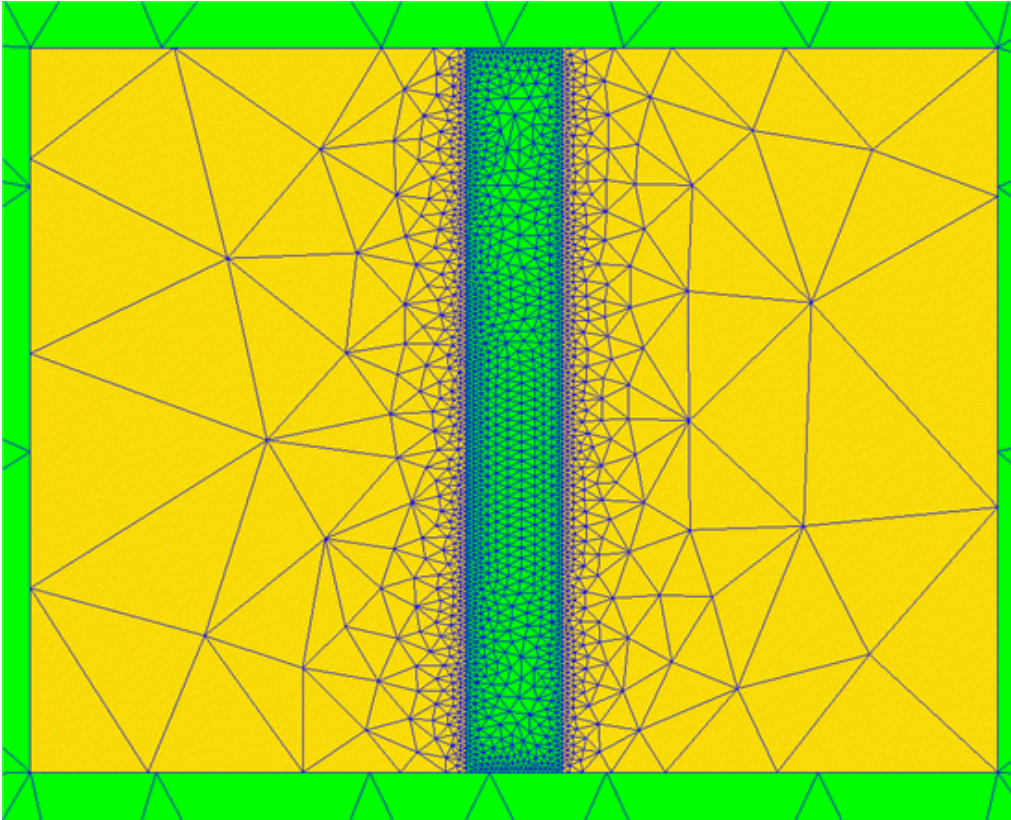
## Meshing Properties Editor

You can select a mesh by using the Meshing Properties Editor. To open this editor, right-click an object and select **Gridding/Meshing > Meshing Properties**.



## Edge Meshing

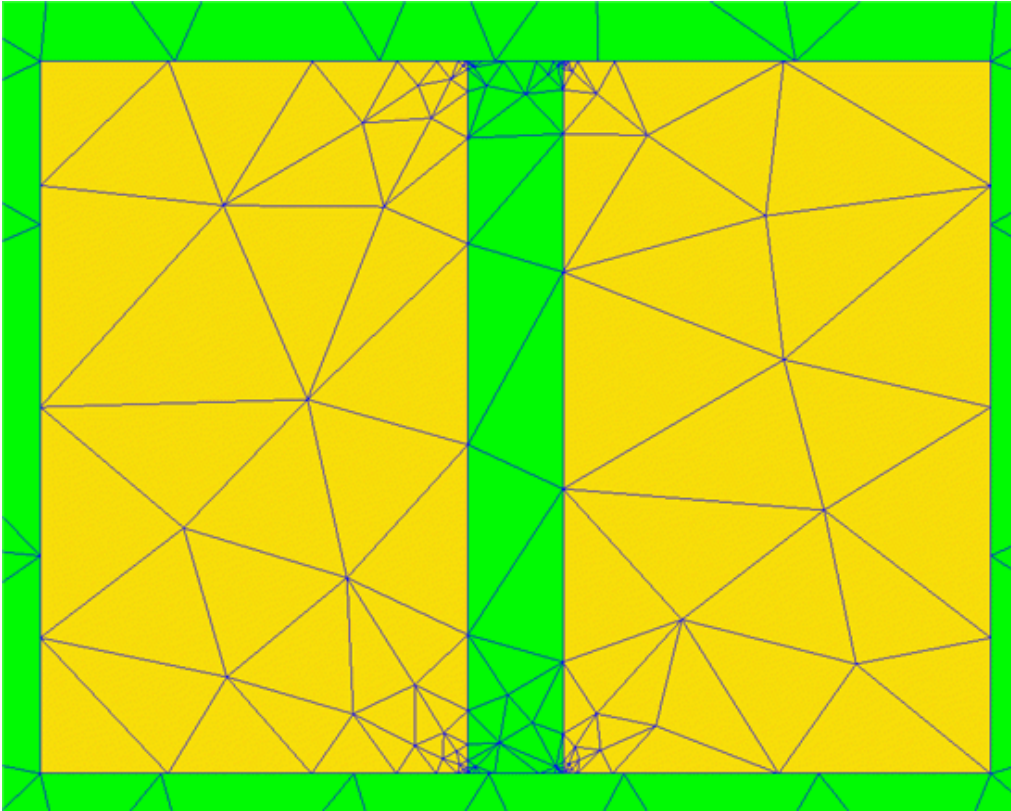
Edge meshing reduces the mesh length around the edges. In the following figure, the edge mesh length is specified as 0.1 mm and mesh is generated by applying an edge mesh to the strip line running on the substrate:



In this figure, around the edges of the strip, the length of mesh edge is reduced to 0.1 mm. A fan-out effect is created to maintain the tetrahedron quality. The edges do not need to be straight lines, any shape is allowed. A good value is one fifth of the width of a strip.

## Vertex Meshing

Vertex meshing reduces the mesh length around vertices. In the following figure, the vertex mesh length is set to 0.033 mm:



When combined with edge meshing, the shortest length of both meshes is chosen in any given region. The region where the value is applied by the vertex mesh length setting is roughly twice its length in each direction: the actual region is still being optimized. The region affected by it depends on local geometry.

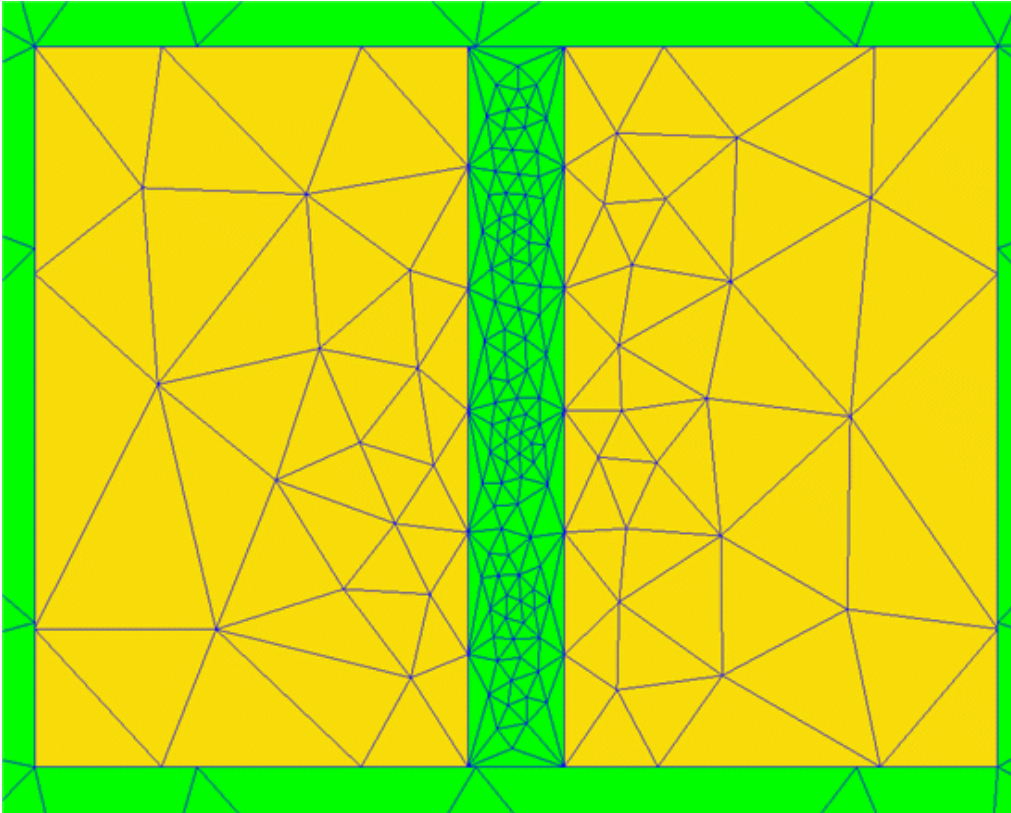
**NOTE**

Vertices in EMPro are zero-dimensional entities, like edges are one-dimensional. You can recognize vertices in straight line designs as the corners of shapes. During sketching vertices will advertise themselves as vertex0, vertex1, when hovering over them.

## Surface Meshing

Surface meshing reduces the mesh length on surfaces of objects. In the following figure, the surface mesh length is set to 0.5 mm, but the mesh length is not exactly 0.5 mm everywhere on the strip:

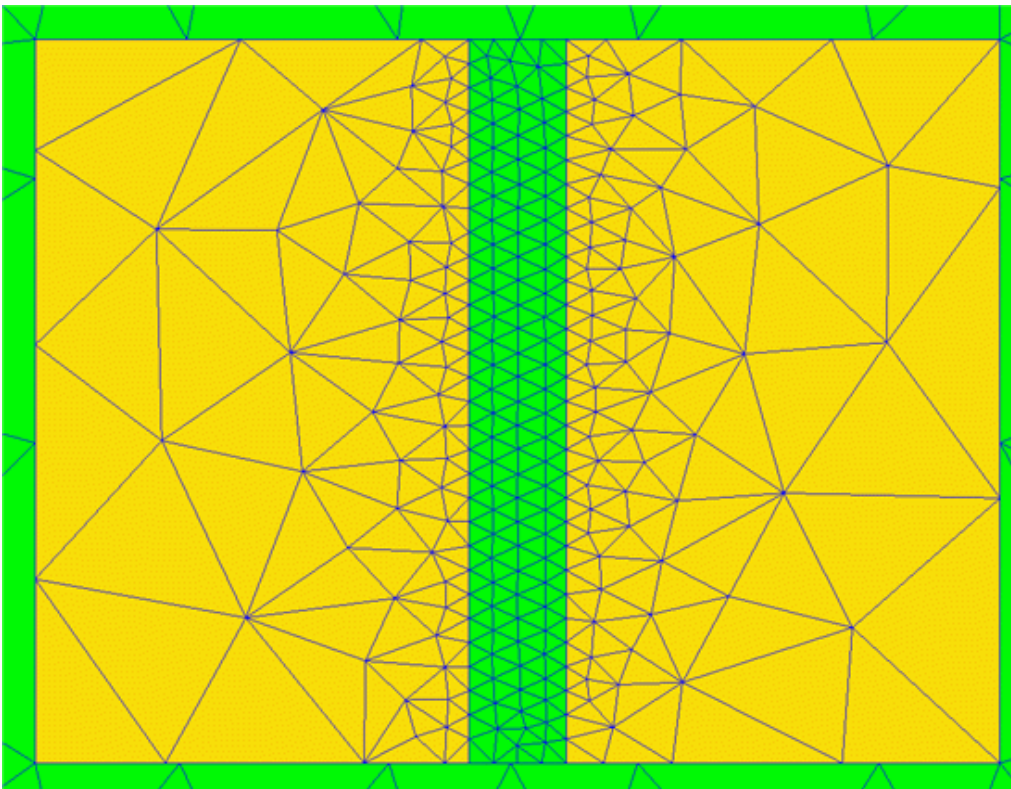




This effect is caused by the fact that the edge mesh length was not explicitly set. In that case, internally, the edge mesh length is reduced to five times the surface mesh length. This is necessary to maintain mesh quality and keep the performance of the meshing process under control. This option is useful when the affected region of the mesh control settings needs to be limited. The surface mesh length is not affecting other objects nearby the strip.

## Combined Edge and Surface Meshing

In the following figure, the edge and surface mesh length have both been set to 0.5 mm:



The main difference with the surface meshing applied in isolation is that the fan out of the meshing reaches into the substrate.

## General Settings and GUI

Edge and vertex mesh length can be set globally and are part of the simulation settings. Objects settings take higher priority, but only when they dictate a smaller mesh length. It is not possible to override an edge length per object with a length larger than the globally applied settings. A value of 0 means that the setting is ignored. Negative values will result in an invalid simulation setup. The values may be parameters if desirable. The per object settings are reachable through the context menu and are part of the meshing properties of an object.

### NOTE

You can edit the mesh settings of multiple objects simultaneously by copying the required settings on objects. In the Meshing Properties Editor, you can use **Copy to Clipboard** button. After the copy operation is done, it is possible to select a group of parts and then request the context menu and perform the “Paste” action (or using the shortcut).

## Edge and Vertex Meshing Guidelines

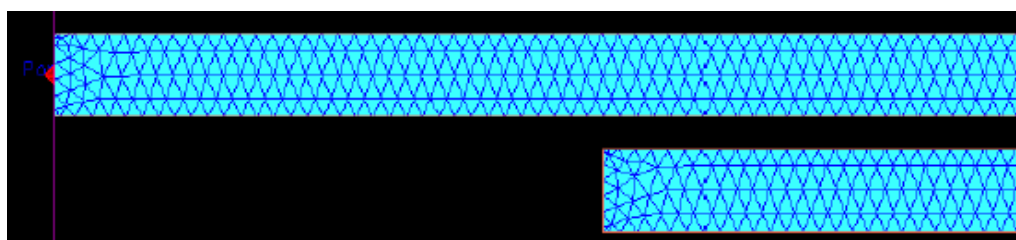
You can refer the following guidelines while applying the edge and vertex meshing:

- For conductive paths, apply an edge mesh with length roughly 1/5th of the width of the strip.
- Use vertex meshing as the additional number of tetrahedra is often quite minimal and the results are more accurate.
- If it is deemed overkill to apply the edge meshing on all conductors, apply at least to the objects closest to a port. For coaxial structures selecting the objects having a face in the plane of the waveguide and selecting an edge mesh there is beneficial.

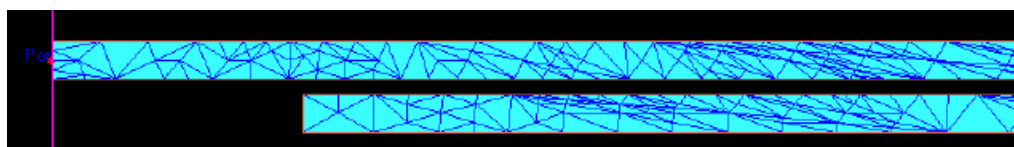
## Adaptive vs Initial Meshing

When advanced mesh controls are used, EMPro generates a large increase in the number of tetrahedra of the initial mesh, although the final mesh typically has the same number of unknowns. Even in that case, it is beneficial to use (for instance) edge meshing. The resulting mesh is always of higher of quality and this influences the quality of the results as well.

The following figure displays the final mesh with Edge meshing on the initial mesh. It displays a converged mesh having edge meshing on conductors applied.



In the following figure, the adaptive refinement process has chosen the points for mesh:



**See Also**

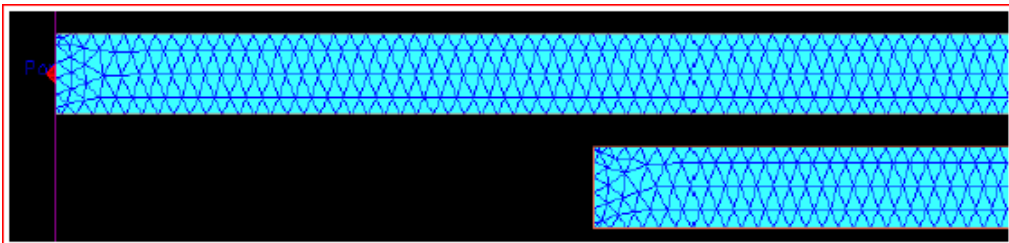
[Specifying FEM Simulation Setup](#)

## Guidelines for Specifying Mesh Settings

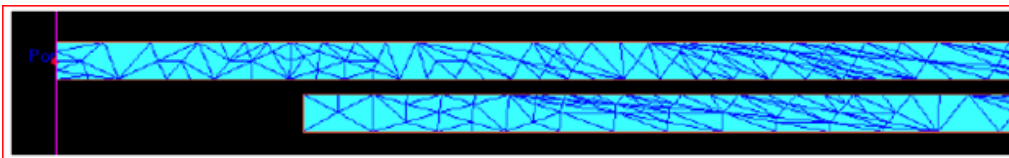
### Adaptive vs Initial Meshing

When advanced mesh controls are used, EMPro can generate an initial mesh with more tetrahedra, although the final mesh typically has the same or less number of unknowns. Even in that case, it is beneficial to use (for instance) edge meshing. The resulting mesh is always of higher quality and this influences the quality of the results as well.

The following figure displays the final mesh with Edge meshing on the initial mesh. It displays a converged mesh having edge meshing on conductors applied.



In the following figure, the adaptive refinement process has chosen the points for mesh:



### Target mesh size

The Automatic Target Mesh size is a good choice for most applications.

For strong radiating designs, it can be advantageous to set a smaller Custom Target Mesh Size: for example, one third of the wavelength at which the radiation occurs. For these radiating designs a second order discretization is recommended. If first order discretization is used further reducing the custom target mesh size to one fifth of the wavelength of radiation produces more accurate results.

Use the Minimal Memory setting for non-radiating designs. The initial mesh is primarily determined by the shapes of the design itself instead of a wavelength. This is useful for designs where the complexity of the shapes of the designs is ensuring dense enough starting meshes in the regions where the electromagnetic fields are concentrated. A set of thick conductors running through the design where the thickness is smaller than the width of the conductors is a typical example.

Common designs sharing these traits are package and module designs.



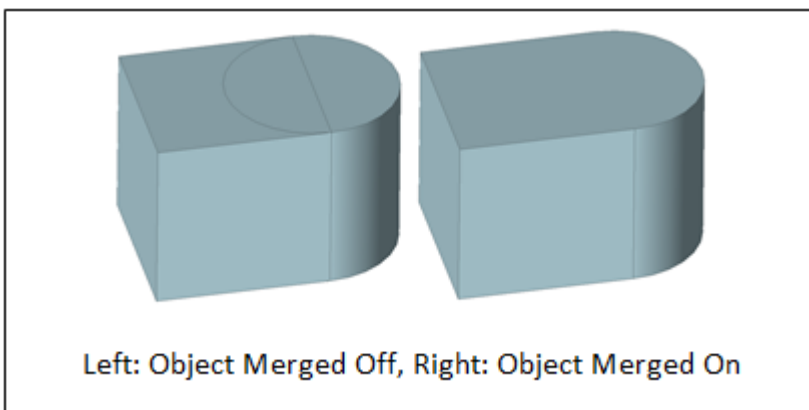
## Minimal Mesh Size

The Minimal Mesh Size Setting located on the advanced properties are used to prevent the meshing process from generating too small tetrahedra. This situation can arise close to curved surfaces where the meshing process potentially over-meshes the structure. When this happens you should use a value that is small enough to describe the curved shape: e.g. choose a distance value that is equivalent to 15-45 degree resolution of an arc or curved surface.

Another situation where the minimal mesh size can be used is when you like to continue using the automatic conductor mesh setting but on some locations the estimated width leads to an unnecessary dense mesh.

## Merge Objects with Same Material

Merging objects with the same material takes all objects that are assigned exactly the same material and share the same mesh priority and lumps them together in one object. Any seam lines that would arise from making the union of these objects are removed. The figure below shows the effect of merging of objects:



By default objects merged with same material is switched on. Using this option will typically give a smaller mesh with similar EM performance. Only in cases where it is deemed mandatory to keep seam lines between objects in the final mesh the option should be turned off. A possible use for this is to force objects lines to be in the mesh for seeding purposes.

## Using Conductor Meshing

Using the Conductor Meshing you can control the density of the initial meshes close to conductors, e.g. to create a denser Edge Mesh type mesh there where you can expect a dense mesh is needed to get an accurate simulation. Using Conductor meshing to seed the initial mesh is useful to speed up the adaptive mesh refinement process.

You can use the Automatic conductor setting when the location of the pins of the port allow to get a good estimate on the edge and vertex values that would be required to obtain an accurate simulation. The estimated widths are reported by the simulator in the log window. Note that this setting will apply the edge to all conductors that are electrically connected at DC to the port (net). Any metal not connected to the port will not get this mesh seeding.

For the designs where metal not electrically connected to the ports is important for the electrical behavior, like a coupled line filter, the Automatic setting is not advised. In case multiple ports are connected to the same net the smallest estimate will be chosen.

The Fixed for all conductors setting applies the specified Edge or Vertex setting to all the metal shapes in the designs which can be overkill in more complex designs. Applying the setting to specific models can be used to be more selective in applying the conductor mesh setting.

## Choosing the Edge and Vertex mesh values

For microwave applications, like a coupled line filter, you can set the edge mesh value to  $1/5^{\text{th}}$  of the width of the strip.

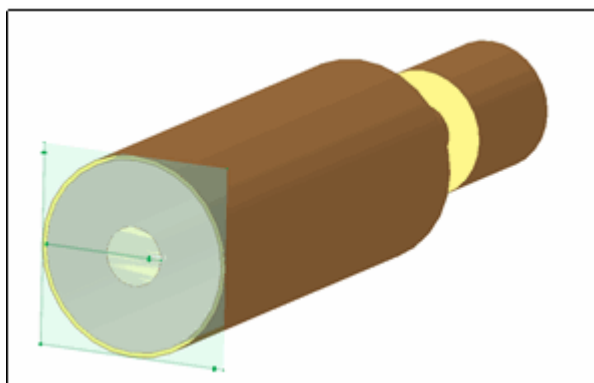
For RF-type of applications, such as RF connectors, modules and packages, you can set the edge mesh value to  $1/3^{\text{th}}$  of the width of the strip.

Use vertex meshing to generate a locally denser mesh around vertices. The region in which the mesh is denser is limited and the impact on the total number of tetrahedral is often quite minimal. The results are that the accuracy of the inductance computation along the conductive paths will typically be more accurate.

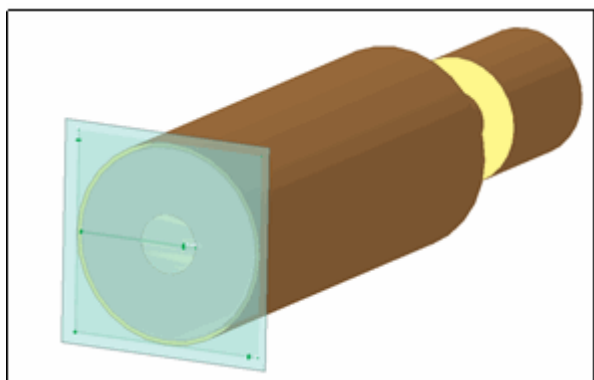
If it is deemed overkill to apply the edge meshing on all conductors, apply the edge meshing at the minimum to the objects closest to a port.

## Coaxial Structures Fed by a Waveguide

Having a good mesh on a waveguide is important to become an accurate simulation result. You can increase the density of mesh on a waveguide by using following technique. Take the coax below.

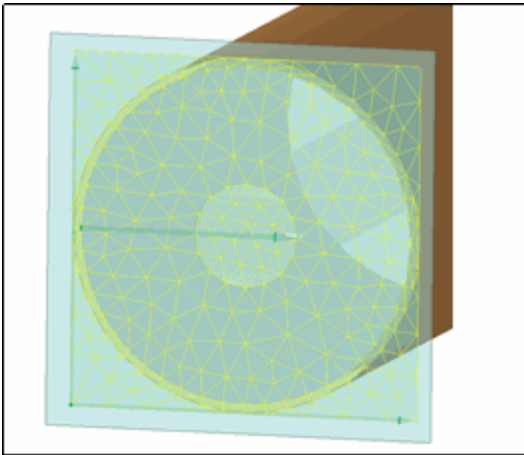


Start by adding a sheet located on the plane of the waveguide and assign it the material air:



Select the sheet and apply mesh settings to generate a mesh of about 200 elements. Do this by setting the edge and surface mesh length to be about  $1/10^{\text{th}}$  of the length of one of the sides of the created sheet. An example setting is shown below. Take note to lower the priority of the meshed object such that it will not overwrite any objects. The sheet object does have to be included in the mesh to have effect though.

The mesh on the port will look as follows:



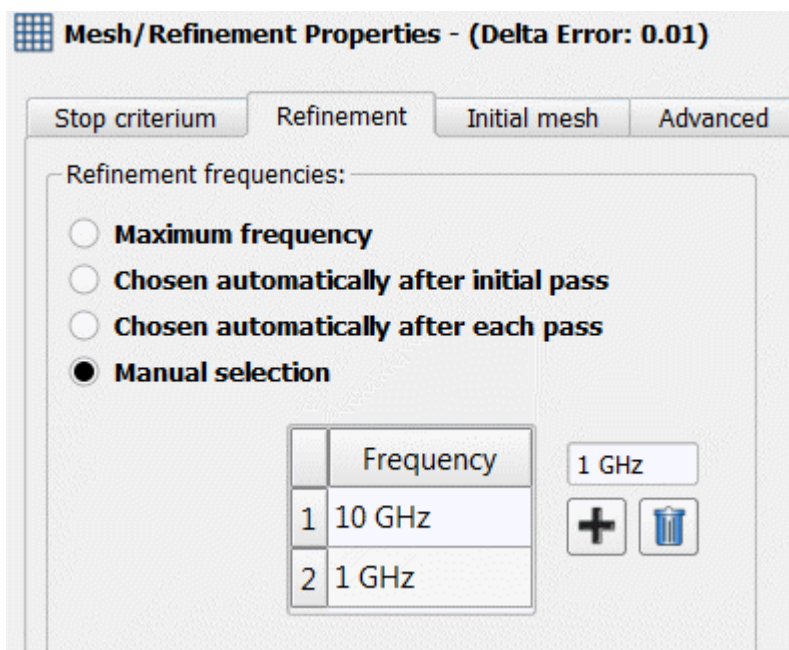
Applying a mesh setting on a conductive net can be done in the platform by using the connectivity tool to select a set of connected parts and then to apply the settings through the clipboard technique.

## Performing FEM Broadband Refinement

You can automate the FEM broadband refinement process. You can select one of the following refinement strategies depending on your requirements:

- Multiple fixed refinement frequencies
- Refinement frequencies chosen automatically after the initial pass
- Refinement frequencies chosen automatically after each pass
- Manual refinement frequency

You can select a refinement strategy from the Setup FEM Simulation window, as shown in the following figure:



This section describes these refinement frequencies and provides guidelines for choosing a strategy.

## Importance of Refinement Frequencies

Mesh refinement procedures in FEM typically consist of the following steps:

1. Simulate at the refinement frequencies.
2. If the calculated S-parameters did not reach convergence:
  - Estimate the discretization errors of the field solution.
  - Refine the mesh at locations with high values for the error.
  - Repeat step 1.

In this procedure, the refinement frequencies play a key role:

- **Convergence monitoring:** the convergence of the refinement process is monitored at the refinement frequencies (step 2).
- **Field error estimation:** the field error estimates, upon which the mesh refinement will be based, are calculated at the refinement frequencies (step 2a).

Consequently, the choice of the refinement frequencies can have an important impact on the accuracy of the simulation results:

- The mesh convergence should be monitored at frequencies where the S-parameters change most during mesh refinement. If this is not the case, the refinement process might stop too early, i.e., *before* the mesh refinement has reached convergence over the entire frequency range of interest.
- All important electromagnetic phenomena should occur at one of the refinement frequencies. If this is not the case, these phenomena cannot be taken into account during mesh refinement. As a consequence, it is possible that the final mesh is not suited to represent these phenomena accurately.

## Performing Manual Broadband Refinement

Previously, FEM mesh refinement considered only one single frequency, which remained fixed during the entire refinement process. Refinement at multiple frequencies could be achieved only manually by *reusing* previous simulation results. For instance, refining at two frequencies  $f_1$  and  $f_2$  is achieved as follows:

1. Simulate with the refinement and solve frequency set to  $F_1$ .
2. Resimulate by using the previous simulations results with the refinement frequency set to  $f_2$ . Also, set the solve frequency plan to the required frequency range.

## Performing Automatic Broadband Refinement

You can select one of the following refinement strategy:

- Multiple Fixed Refinement Frequencies
- Refinement Frequencies chosen Automatically After Initial Pass
- Refinement Frequencies Chosen Automatically After Each Pass

## Multiple Fixed Refinement Frequencies

The user can specify a set of refinement frequencies by selecting the button 'Manual selection', and adding frequencies to the frequency table. This refinement strategy behaves as follows:

- All refinement frequencies will be simulated on *each* mesh level
- Convergence is reached if the convergence criterium is satisfied for *all* refinement frequencies. In other words, convergence depends on the *worst case refinement frequency* on each level.
- The mesh refinement will depend on the error estimates of *all* refinement frequencies.

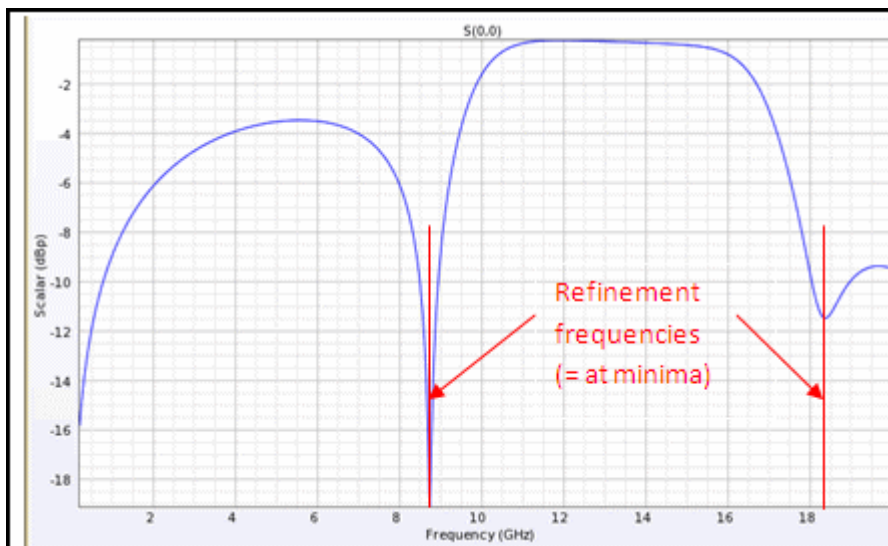
The corresponding logfile shows:

- Delta(S) for each simulated frequency
- Worst case Delta(S) for each mesh level (last line)

REFINING										
level	freq(GHz)	MESHING			SOLVING					DeltaS
		alTolr	Elapsed time	CPU time	nNodes	mem(MB)	Elapsed time	CPU time		
1	2.000	272	00:00:02.2	00:00:01.1	1848	0.100	00:00:00.8	00:00:00.9	/	
	6.000					0.100	00:00:00.1	00:00:00.2	/	
	8.000					0.100	00:00:00.0	00:00:00.1	/	
	(8.000)									
2	2.000	512	00:00:00.4	00:00:00.3	3478	0.108	00:00:01.1	00:00:01.4	0.607[->0.010]	
	6.000					0.108	00:00:00.1	00:00:00.2	0.620[->0.010]	
	8.000					0.108	00:00:00.1	00:00:00.2	0.626[->0.010]	
	(8.000)								0.626[->0.010]	
3	2.000	1219	00:00:00.8	00:00:00.7	8136	0.137	00:00:00.7	00:00:01.2	0.607[->0.010]	
	6.000					0.137	00:00:00.2	00:00:00.6	0.620[->0.010]	
	8.000					0.137	00:00:00.2	00:00:00.6	0.625[->0.010]	
	(8.000)								0.625[->0.010]	
4	2.000	1969	00:00:01.3	00:00:01.2	13048	0.164	00:00:00.9	00:00:02.0	0.603[->0.010]	
	6.000					0.164	00:00:00.3	00:00:00.9	0.609[->0.010]	
	8.000					0.164	00:00:00.3	00:00:01.0	0.612[->0.010]	
	(8.000)								0.612[->0.010]	
Worst case Delta(S)										

## Refinement Frequencies chosen Automatically After Initial Pass

In this refinement strategy, the refinement frequencies are not user-defined, but chosen by the solver. For this purpose, the entire target frequency plan is simulated on the initial mesh. Based on an analysis of the resulting S-parameters, a set of refinement frequencies is chosen. Typically, these frequencies will be the frequencies for which the diagonal elements of the S-parameter matrix reach a minimum. The selected refinement frequencies will be used in all subsequent refinement passes.



Except for the AFS sweep on the initial mesh level, this refinement strategy is identical to the strategy based on user-defined refinement frequencies.

REFINING										
Level	freq(GHz)	MESHING			SOLVING					
		nbTets	Elapsed time	CPU time	nbUnknowns	mem(GB)	Elapsed time	CPU time	Delta(S)	n
1	0.200	8501	00:00:09.4	00:00:08.4	54582	0.596	00:00:03.4	00:00:07.3	/	
	20.000					0.596	00:00:01.5	00:00:04.9	/	
	13.400					0.596	00:00:01.5	00:00:04.9	/	
	6.800					0.596	00:00:01.5	00:00:04.9	/	
	3.500					0.596	00:00:01.5	00:00:04.9	/	
	10.100					0.596	00:00:01.5	00:00:05.0	/	
	8.450					0.596	00:00:01.6	00:00:05.0	/	
	11.750					0.596	00:00:01.5	00:00:04.8	/	
	17.800					0.596	00:00:01.5	00:00:05.0	/	
	16.333					0.596	00:00:01.5	00:00:05.4	/	
	18.900					0.596	00:00:01.5	00:00:05.0	/	
	18.350					0.596	00:00:01.5	00:00:05.3	/	
2	8.800	10524	00:00:04.7	00:00:04.5	67852	0.686	00:00:03.8	00:00:05.0	0.026 [->0.010]	
	18.300					0.686	00:00:02.1	00:00:06.9	0.026 [->0.010]	
	(18.300)								0.026 [->0.010]	
3	8.800	12516	00:00:05.9	00:00:05.7	88956	0.772	00:00:07.7	00:00:11.9	0.089 [->0.010]	
	18.300					0.772	00:00:02.8	00:00:05.6	0.085 [->0.010]	
	(8.800)								0.089 [->0.010]	

AFS sweep on initial level used to determine the refinement frequencies

Fixed refinement frequencies

## Refinement Frequencies Chosen Automatically After Each Pass

In this refinement strategy, the refinement frequencies are not fixed, but can vary from level to level. This refinement strategy behaves as follows:

- The entire target frequency plan will be simulated *on each mesh level using an adaptive frequency sweep*.
- The difference of the S-parameters of the *rational models will be monitored*, instead of the difference between the S-parameters of the simulated frequencies only. In other words, convergence depends on the worst case frequency in the entire frequency range of interest. This is the most detailed convergence monitoring possible.
- The mesh refinement will depend on the error estimates of *all simulated frequencies*.

REFINING										
level	freq(GHz)	MESHING			SOLVING					
		nbTets	Elapsed time	CPU time	nbUnknowns	mem(GB)	Elapsed time	CPU time	Delta(S)	n
1	0.200	8501	00:00:09.2	00:00:08.2	54582	0.658	00:00:02.9	00:00:06.6	/	
	20.000					0.658	00:00:01.5	00:00:04.8	/	
	13.400					0.658	00:00:01.5	00:00:04.7	/	
	6.800					0.658	00:00:01.5	00:00:04.8	/	
	3.500					0.658	00:00:01.5	00:00:05.0	/	
	10.100					0.658	00:00:01.5	00:00:04.7	/	
	8.450					0.658	00:00:01.5	00:00:04.9	/	
	11.750					0.658	00:00:01.5	00:00:05.0	/	
	17.800					0.658	00:00:01.5	00:00:04.9	/	
	16.333					0.658	00:00:01.5	00:00:05.2	/	
	18.900					0.658	00:00:01.5	00:00:05.4	/	
	18.350					0.658	00:00:01.5	00:00:05.2	/	
2	0.200	10524	00:00:04.6	00:00:04.5	67852	0.627	00:00:09.3	00:00:09.4	1.5e-04 [->0.010]	
	20.000					0.627	00:00:02.0	00:00:06.8	0.015 [->0.010]	
	13.400					0.627	00:00:02.0	00:00:07.1	0.019 [->0.010]	
	6.800					0.627	00:00:02.0	00:00:07.0	0.005 [->0.010]	
	3.500					0.627	00:00:02.0	00:00:06.8	0.002 [->0.010]	
	10.100					0.627	00:00:02.0	00:00:06.8	0.034 [->0.010]	
	8.450					0.627	00:00:02.0	00:00:06.9	0.015 [->0.010]	
	11.750					0.627	00:00:02.1	00:00:06.8	0.021 [->0.010]	
	17.800					0.627	00:00:02.1	00:00:07.0	0.037 [->0.010]	
	16.333					0.629	00:00:02.0	00:00:06.9	0.057 [->0.010]	
	18.900					0.629	00:00:02.1	00:00:06.9	0.020 [->0.010]	
	18.350					0.629	00:00:02.1	00:00:07.2	0.025 [->0.010]	
3	(16.670)								0.060 [->0.010]	

Worst case frequency and corresponding Delta(S)

## Selecting a Refinement Strategy

You can refer the following guidelines while selecting a refinement strategy:


- **Multiple fixed refinement frequencies:** This strategy computes the fewest frequency points during the adaptive refinement stage and therefore will usually lead to the fastest simulations. This strategy is recommended for:
  - Electrically short transition structures. These can include matching networks, packages, impedance transformations, etc. If the transmission parameters accumulate less than 180 degrees of phase, then it is sufficient to refine only at the highest frequency.
  - Designs where you know the frequencies of the critical features. These can include dual (or multi-) band antennas, band pass filters etc. For these structures set the refinement frequencies to the frequencies of most interest (i.e. in each antenna frequency band, at the low end and high end of the band pass filter). In many cases it is advisable to also include the maximum frequency of the simulation as a refinement frequency.
  - Initial analysis of a structure where speed is preferred over accuracy.
- **Refinement frequencies chosen automatically after the initial pass:** This strategy is a compromise between manually selecting the refinement frequencies and letting the FEM simulator automatically select the refinement frequencies at each adaptive pass. This strategy is recommended when you do not have a good idea a priori where the important characteristics will occur in your frequency but you know the frequency transitions are not extremely sharp (i.e. are not high Q resonances).
- **Refinement frequencies chosen automatically after each pass:** This is the automated strategy and tends to generate accurate results, but it also tends to result in the longest simulation times. This strategy is recommended:
  - For structures with high Q resonances.
  - During final analysis of a structure where accuracy is preferred over fast simulations.

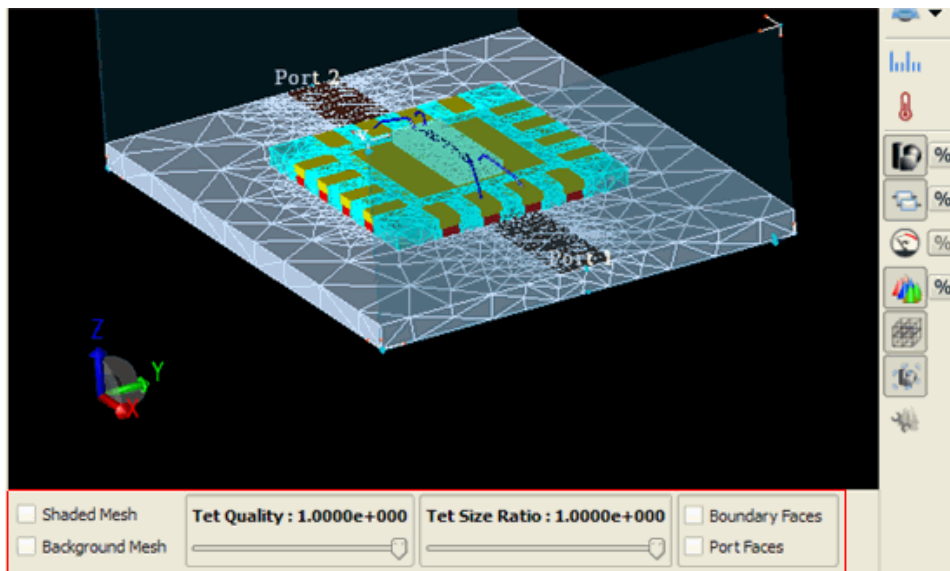
## Viewing FEM Mesh

You can display the following types of mesh view in an FEM simulation:

- Shaded mesh
- Background mesh
- Boundary mesh
- Waveguide ports mesh

To view a mesh in FEM simulations:

1. Run an FEM simulation.
2. Click **Toggle Mesh Viewing Controls** (  ) in **View Tools**. The mesh that is displayed is the one that was used in the last successful FEM simulation. If an FEM simulation did not complete successfully, its mesh will not be available for viewing.
3. The mesh controlling options are displayed at the bottom of the Geometry window, as shown highlighted in the following figure:



Using the mesh view control options, you can perform the following tasks:

- Controlling the Mesh View
- Viewing a Shaded Mesh
- Viewing a Background Mesh
- Viewing a Mesh for Boundary Faces
- Viewing a Mesh for Port Faces

## Show me How Do I View Mesh



Video: How to View FEM Mesh

### NOTE

You should generate a valid mesh before using the mesh viewing options.

## Controlling the Mesh View

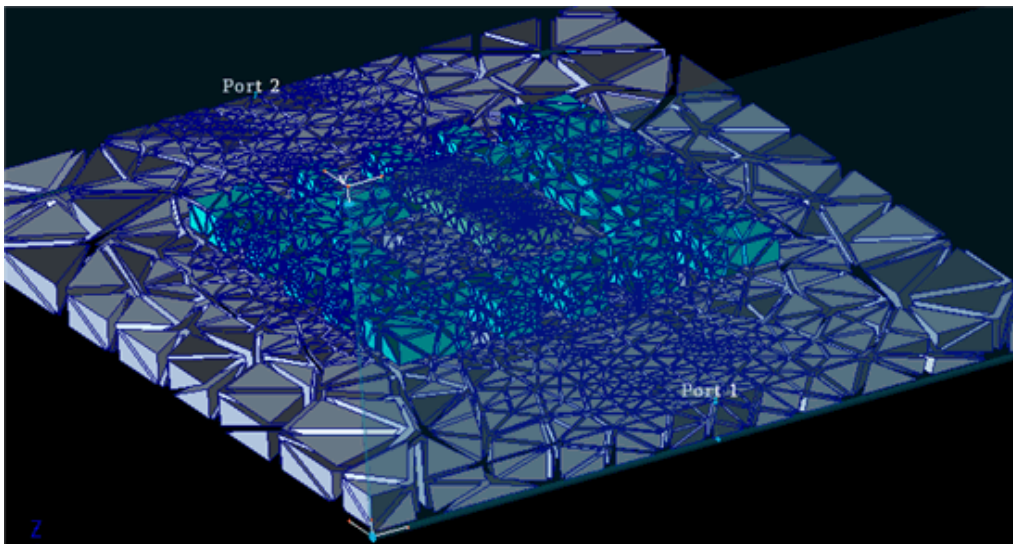
The mesh control bar contains two slider bars that allow you to control the relative quality or relative size of the tetrahedra being displayed. Both quantities are relative numbers. The quality of a tetrahedron reflects that its **flat** structure. If a tetrahedron is flat, it is more difficult to get an accurate field representation in that tetrahedron. The volume control operates in a similar manner, except that it uses the relative size of the tetrahedrons as a measure.

It is also possible to control the visibility of the mesh by using the Material Menu. Right-click a material to select the **Show FEM Mesh** and **Hide FEM Mesh** options, which allow you to turn on or off the display of the mesh in that material.

## Viewing a Shaded Mesh

To view a shaded mesh, select the **Shaded Mesh** option. The following figure displays a shaded mesh view:

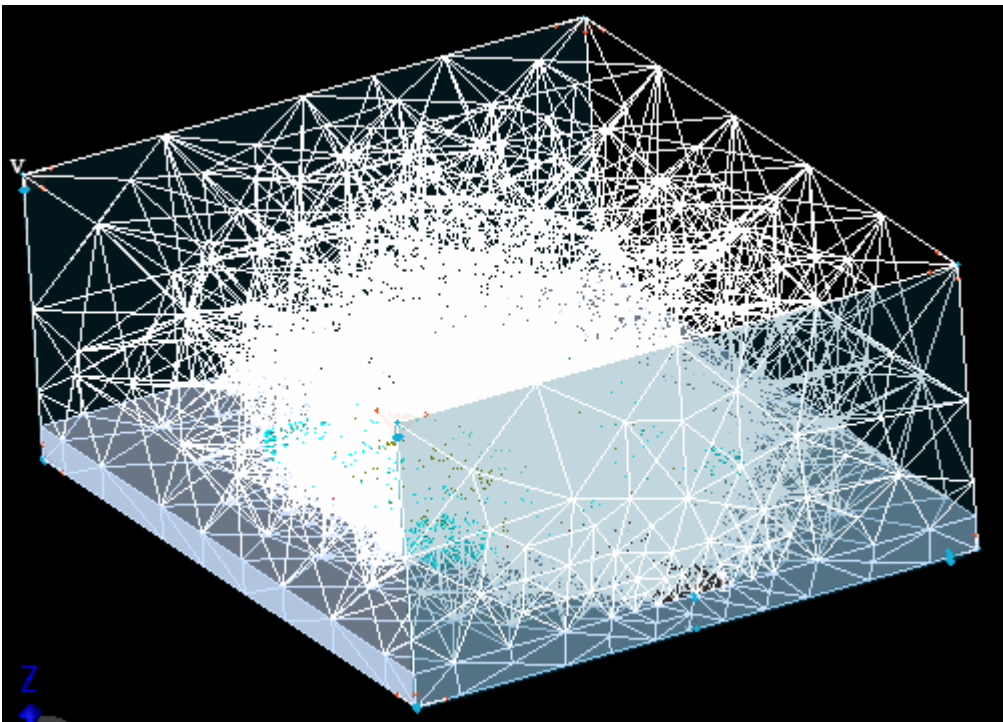


**NOTE**

Remove selection from the **Toggle Parts Visibility** option to view a shaded mesh.

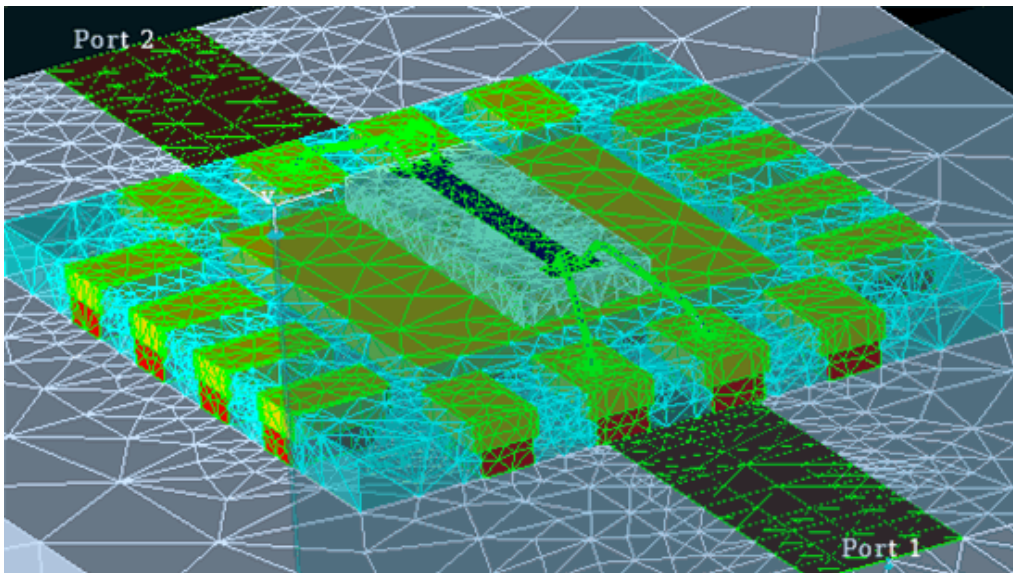
## Viewing a Background Mesh

To view the background mesh for free space, Select **Background Mesh**. The following figure displays a mesh view for free space in the background:



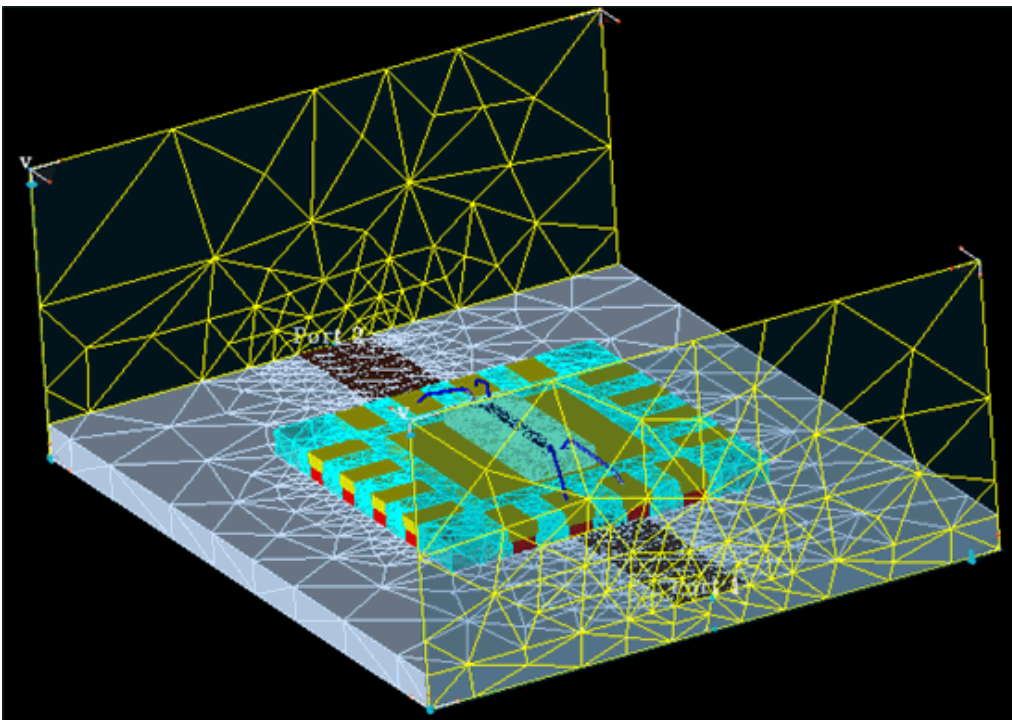
## Viewing a Mesh for Boundary Faces

To display a mesh view for boundary faces, select **Boundary Faces**. The following figure displays a mesh view for boundary faces:



## Viewing a Mesh for Port Faces

To display a mesh view for waveguide ports, select **Port Faces**. The following figure displays a mesh view of waveguide ports:




## Reusing Mesh and Frequency Points

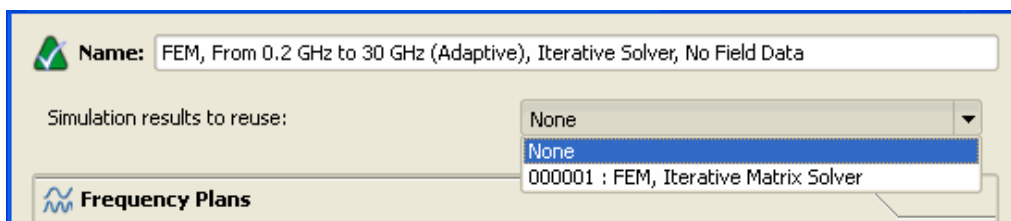
EMPro enables you to reuse any existing FEM simulation. For a new FEM simulation, you can use the existing simulation settings for mesh refinement, AFS points, and solver. You can also use the results from an existing FEM simulation in a new FEM simulation. In this case, results from the existing simulation will be the starting point for the new simulation.

## Using the Existing Simulation Setup Settings

To demonstrate how to reuse the simulation setup, open the QFN Package project from the example set:

1. Unarchive the **QFN Package** project from **Help > Example**.
2. Open the *QFN Package.ep*.
3. Click **Simulations**.
4. Click **Setup** (  ) to create and edit the setup of a new simulation. The **Setup FEM Simulation** window is displayed.
5. Add the Frequency Plans relevant for your project.
6. Click **Create and Queue Simulation**.

You can reuse this simulation in another simulation by selecting a simulation from list of already solved simulations from the **Simulation Results to Reuse** drop-down list. Change the pre-existing simulation settings according to your new requirements, and start the new simulation. In the output, it will be shown which results are reused.



You can reuse simulations in the following scenarios:

- **Increasing the maximum number of refinement steps:** This can be useful if the mesh refinement did not converge.
- **Increasing the accuracy:** This can be done by decreasing target delta S or by increasing the number of consecutive passes the target delta S criterium must be satisfied. This will result in additional refinement steps.
- **Changing the refinement frequency:** Additional refinement steps will be done for the new refinement frequency. This can be useful if the field solutions are fundamentally different for different frequencies. It also can be used to achieve a broadband refinement.
- **Increasing the maximum number of frequency points for AFS:** This is useful if AFS did not converge.
- **Using a different setting for the linear solver:**
  - If the direct solver aborted because of an out of memory problem, the simulation can be continued by reusing it with the iterative solver selected instead.
  - If the iterative solver aborted because it did not converge, the simulation can be continued by reusing it by allowing more iterations for the iterative solver or decreasing the tolerance for the iterative solver or with the direct solver selected instead.
- **Changing the frequency plan:**
  - Extending the frequency range.
  - Adding discrete frequencies: this can be used to force AFS to zoom in into a specific frequency range.

## Troubleshooting Mesh Failures

If the mesh generation fails in the surface meshing a list of failed faces assembled in an SAT file will be generated in the design directory. The exact location of this file will be shown in the output window of the simulation.

## Best practices

- Simplify the design by removing unnecessary (for the EM-modeling) detail. Various geometry modification tools are available within the EMPro GUI to achieve this.
- Avoid shelled objects where the thickness of the shell is much smaller than the total dimensions of the object. In this case, replace the thick shell with a thin sheet.
- Visually inspect the design for small gaps between two 3D objects. Consider removing the gaps by moving one of the objects to create overlap between objects.
- Avoid using complex Boolean subtractions: use the mesh priority to set the correct object precedence for meshing. For example, a metal object sticking through a dielectric object will be meshed as such if the priority of the metal object is set to a higher value than the dielectric object.

To isolate specific objects that are causing mesh failures, you can selectively include or exclude objects from a simulation by using the Include In Mesh option in the context menu of objects.

## Known Issue

### Using English units may create small gaps or slivers in some designs

EMPro uses SI (metric) units as the default system units. When a design is created and modified using English units, an internal unit conversion is made during each operation. Each internal conversion introduces small rounding errors. As more operations are completed, the rounding errors can accumulate to the point where small gaps or slivers may appear in the design. The FEM mesher has difficulty in meshing geometries containing these small gaps or slivers. This problem can be minimized either by finding and eliminating the small gaps or slivers, by reducing the number of operations on a design (reducing the number of internal unit conversions), by using metric units during the creation and modification of the design (eliminating internal unit conversions), or by using “Match Points” alignment tool. This issue will be addressed in the next release of EMPro.

## Solution

- Import designs using metric units. This will reduce the occurrence of errors caused by the conversion between metric units and English units.
- When designs are imported in English units, move designs as a group, and minimize unnecessary drawing operations to avoid the creation of small gaps or slivers.
- When performing operations, such as moving, on designs containing imported objects, using the “Match Points” alignment tool may eliminate any small geometrical offsets created.
- Create parametric equations in metric units. The conversion between metric and English units has limited precision, so mixing parametric equations and direct coordinate entry may introduce small gaps or slivers.

# Using Ports in FEM Simulation

---

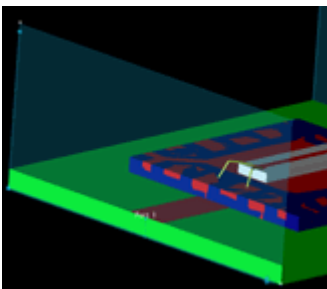
Port definitions are the connection between EM simulation and circuit simulation. This means that where you define the ports (and where you introduce voltages and currents), the conditions of a valid circuit simulation setup need to apply. This points directly to the electrical sizes between nodes that are connected to a circuit component or to a port having to be small compared to the wavelength (ideally zero). It is recommended to be smaller than a tenth of a wavelength.

FEM supports the following types of ports:

## Waveguide Ports (2D-plane source)

---

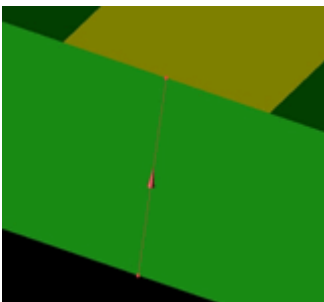
- Calibrated 2D-plane source
- User defines a 2D plane for the port in the simulation domain with a source impedance, which could be simply 50 ohms or RLC
- Higher order modes calculations are available
- Most accurate
- Must be placed outside 3D structure



## Internal ports

---

- Voltage / Current 'line' sources that can be placed inside 3D structure
- Include parasitic inductance that may reduce simulation accuracy



## Internal sheet ports

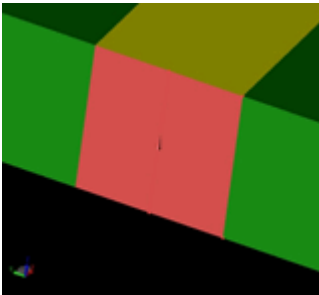
---

- Voltage / Current planar sources that can be placed inside the 3D structure

- Reduced parasitic inductance compared to standard internal port

**NOTE**

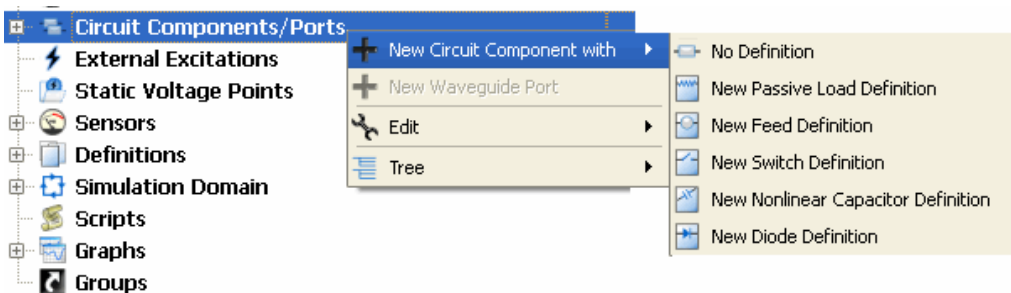
A sheet port needs to align with the polarity of a port. Otherwise, the sheet port becomes twisted.



You can define these ports without defining a new feed definition or with a new feed definition.

## Circuit Component Port

EMPro supports the **Component Feed** port in the FEM simulation setup. The Component Feed ports are direct point feeds having a negative and positive connection point to the metallization of the design.



FEM supports only the **New feed** definition. The Passive load, Switch, Non Linear Capacitor and Diode ports are not supported in FEM. For more information about Component ports, refer to the [Adding Circuit Components](#) section.

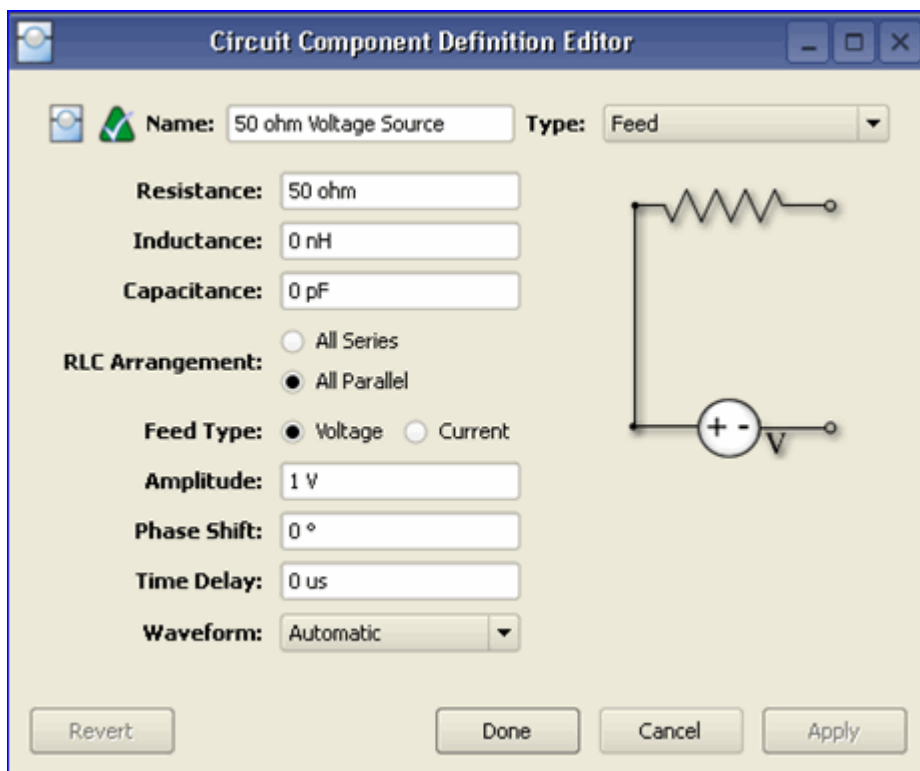
## Component Feeds

EMPro supports the **Component Feed** port in the FEM simulation setup. The Component Feed ports are direct point feeds having a negative and positive connection point to the metallization of the design. For more information about Component Feeds, refer to the [Adding Circuit Components](#) section.

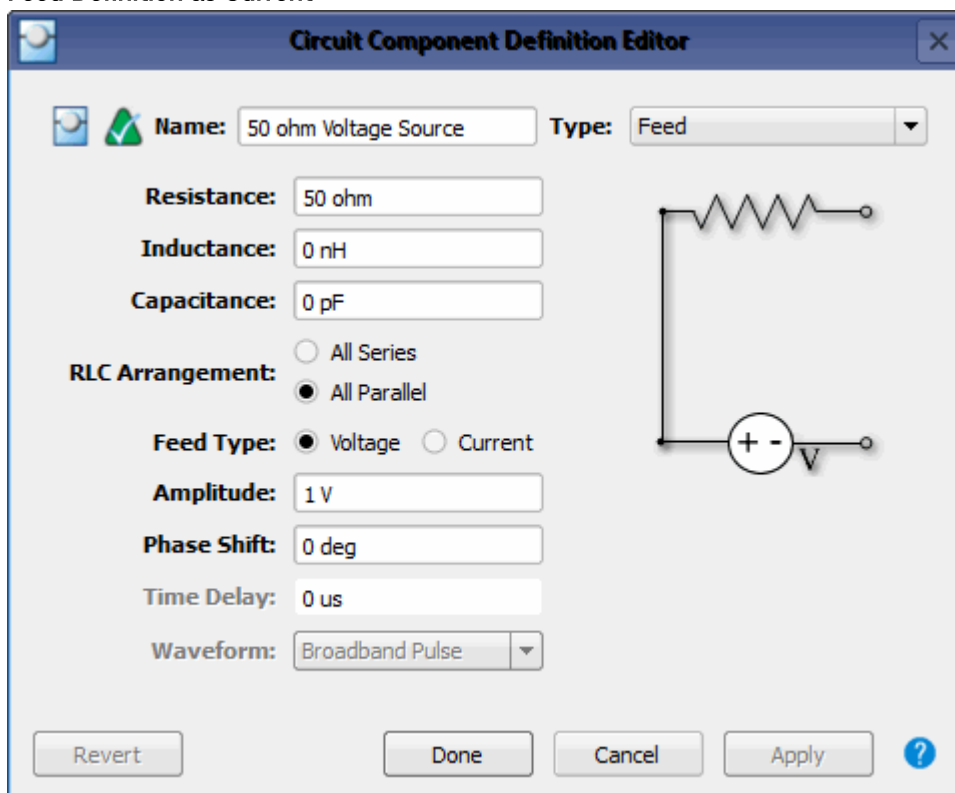
For FEM simulations, only the following component feed definitions are allowed, which specify the voltage sources with a termination impedance.

### Feed Definition as Voltage





#### Feed Definition as Current



The termination resistance can be of any value, but Amplitude can be maximum of 1 V or 1 A. If while defining component port **No definition** option is used, then after defining port, appropriate definition of the Voltage source or Current source must be applied.

# Waveguide Port

Waveguide ports enable you to shift the location of the reference lines used for the S-parameters. It consists of the following features:

- Calibrated 2D-plane source
- User defines a 2D plane for the port in the simulation domain with a source impedance, which could be simply 50 $\Omega$  or RLC
- Higher order modes calculations are available

You can select the following types of reference offsets:

- Specifying a positive reference offset: Shifts the reference planes in the direction towards the circuit.
- Specifying a negative reference offset: Shifts the reference planes in the direction away from the circuit.

## Internal Sheet Ports with Reduced Parasitics

The primary non-compensated (or parasitic) effect of the internal ports implemented in EMPro FEM is a self-inductance generated by the current impressed locally by the source.

In EMPro 2009, this locally impressed current has a distribution that is localized to the impedance line which is the path along which the voltage of the source is computed (displayed in the 3D EM Preview window). The parasitic inductance of this source is approximately the inductance of a wire that has the length of the *impedance line* and a small *effective radius*. The effective radius is impacted by both the local mesh density and mesh orientation.

In EMPro 2011.02, the impressed current of a current source can be distributed uniformly along a user-defined sheet. As such, the parasitic inductance of this source is essentially the inductance of a sheet of uniformly distributed current with an *effective thickness*. This thickness is of the same order as the *effective radius* of the *non-distributed* source implementation.

### NOTE

A sheet port needs to align with the polarity of a port. Otherwise, the sheet port becomes twisted.

Sheet current sources have the following advantages in comparison to non-distributed current sources:

- Lower parasitic inductance.
- Decrease in the current fan-out as the current spreads to be the current distribution of the signal line.

### NOTE

In EMPro 2013.07, an automatic calibration routine is added for FEM circuit sources with a sheet extent to remove the self-inductance of the sheet of current that flows through the source. Circuit sources with “Use Sheet” disabled yet calibrated in FEM (and thus include excess inductance).

### See Also

[Simulating a Microstrip Line with Sheet Port](#)



# Creating Waveguide Ports

---

## Contents

---

- [Setting up Waveguide Ports](#)
- [Specifying Port Dimensions](#)
- [Specifying Port Properties](#)
- [Creating an Impedance Line](#)

## Setting up Waveguide Ports

---

Waveguide ports enable you to shift the location of the reference lines used for the S-parameters. These ports are only available on bounding box of geometry and excite the modal field/current distribution on surface. A Waveguide port is supported only in FEM. It is not supported in FDTD simulations. To activate the Waveguide port, the application should be in FEM mode. They are Inherently calibrated at all frequencies and use the Eigen-mode solver to find N modes with the highest propagation constants (where N is equal to the number of impedance lines).

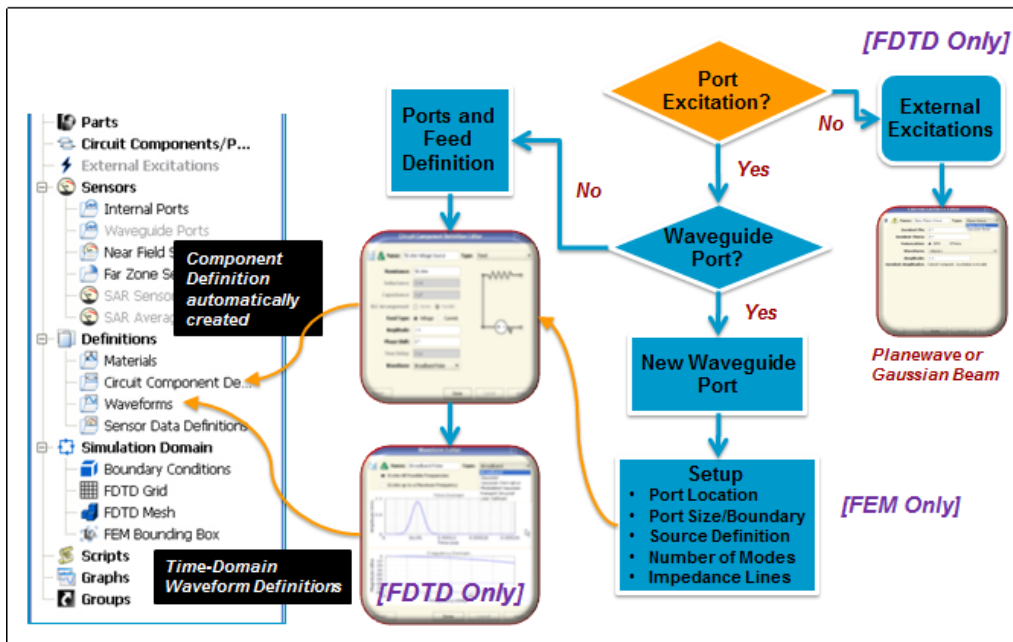
Creating Waveguide Ports or Line Sources automatically creates:

- Sensors to sample the transmitted/reflected energy.
- Circuit Component (Feed) Definitions, such as voltage, current, and power sources.
- Link between feed definitions and ports.

It consists of the following features:

- Calibrated 2D-plane source.
- User defines a 2D plane for the port in the simulation domain with a source impedance, which could be simply 50 ohms or RLC.
- Higher order modes calculations are available.
- Most accurate.
- Must be placed outside 3D structure.

The following figure displays Port/Feed Definition Flow in EMPro:




Adding a port, includes selecting the object face, defining voltage, and defining an impedance line. To create a waveguide port:

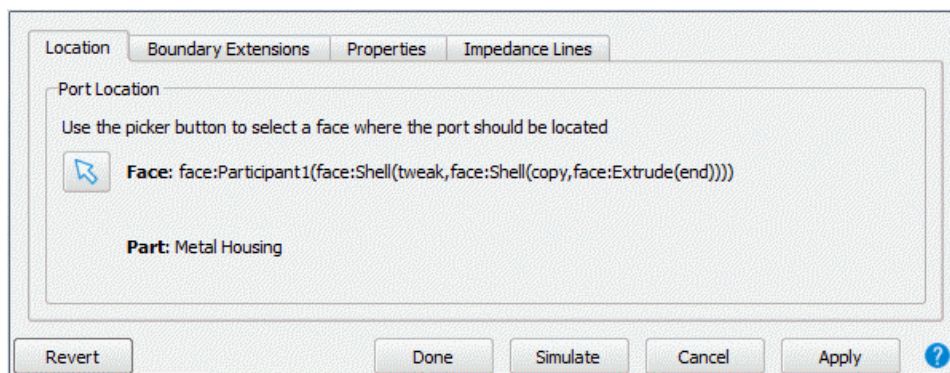
1. Select or choose a 2D face on object where the waveguide port will be located.
2. Specify the size of port by entering numbers for u, v extension from the selected face.
3. Define nodal or modal, and number of modes.
4. Define impedance lines to calculate port impedance.

You can perform all these tasks by using the EMPro Waveguide Ports Editor.



## Creating a Waveguide Port

To create a waveguide port:

1. Right-click **Circuit Components/Ports** and select **New Waveguide Port**. After a new waveguide port option is chosen, the EMPro Waveguide Ports Editor is displayed.
2. Click  to select a 2D face on object where the waveguide port will be located. After you have selected the face, all the tabs are enabled in the EMPro Waveguide Ports Editor.



3. Click the **Editcrosssectionpage** tab.

4. Specify the size of port by entering numbers for u, v extension from the selected face. For more information, see [Specifying Port Dimensions](#).
5. Click the **Properties** tab in the EMPro Waveguide Ports Editor. For more information, see [Specifying Port Properties](#).
  - Type a name for your waveguide port.
  - Select an option from the **Waveguide Port Definition** drop-down list. For more information, see [Waveguide Port Definition](#)
  - Specify the number of modes for this port from the Number of Modes field. For more information, see [Modes](#)
  - Select a value from the **Impedance Definition** drop-down list.
  - Specify a value in the Reference Offset field. For more information, see [Reference Offsets](#)
6. Click the **Impedance Lines** tab. For more information, see [Creating an Impedance Line](#).
  - Select a mode from the **Mode Number** drop-down list.
  - Select the start point of the line by either typing the coordinates in the X, Y, and Z fields or by clicking  in **Endpoint1** to select the X dimensions.
  - Select the end point of the line by either typing the coordinates in the X, Y, and Z fields or by clicking  in **Endpoint2** to select the X dimensions. The endpoint2 values define the direction and length of the impedance line.
7. Click **Done**.

## Guidelines for Waveguide Port Size

For bounded waveguide structures (coax, rectangular waveguide):

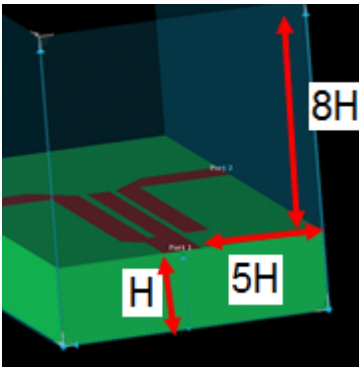
- Waveguide surface needs to completely 'cover' the waveguide, but minimize the overhang.
- Avoid extending surface beyond a ground plane.

For multi-conductor lines:

- Waveguide surface needs to completely 'cover' all 'coupled' conductors of the line (including the reference ground).
- Include ALL conductors in one port with multiple modes than to create separate waveguide surfaces for each signal line.
- Waveguide surfaces cannot overlap.

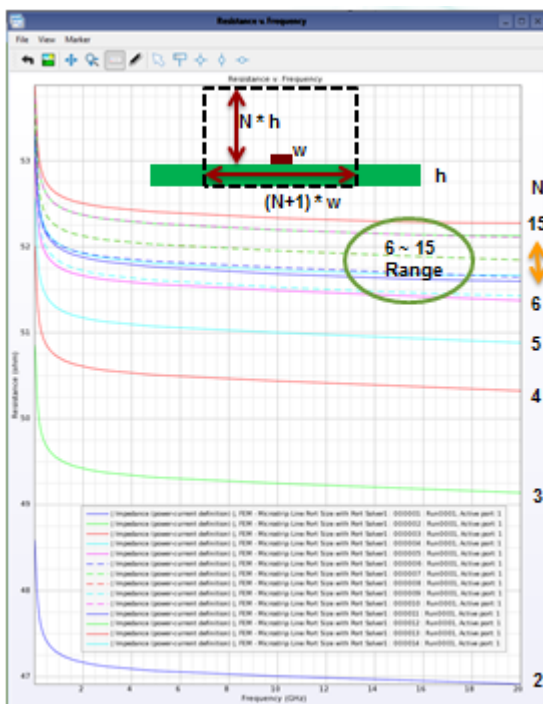
It is recommended to extend a port surface at least three to five times the substrate thickness laterally and eight to ten times the substrate thickness vertically.

Generally, extend the waveguide surface the entire surface of the bounding box face.



## Specifying Port Dimensions

Specifying the port size is important because the port boundary may interact with the structures. Editing a cross section reduces the port size (convert coupled, multi-mode ports into single-mode ports). For example, in the following figure, port impedance of microstrip transmission line versus the size of port dimension plot is displayed:



Recommended port size for few structures

- For bounded waveguide structures (coax, rectangular waveguide), the waveguide surface needs to completely 'cover' the guide, but minimize the overhang. You should avoid extending surface beyond a ground plane.
- For multi-conductor lines, waveguide surface needs to completely 'cover' all 'coupled' conductors of the line (including the reference ground). It is generally better to include ALL conductors in one port with multiple modes than to create separate waveguide surfaces for each signal line.

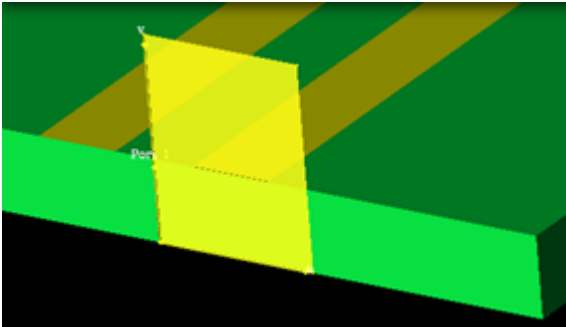
To specify port dimensions:

1. Open the EMPro Waveguide Ports Editor by right-clicking **Circuit Components/Ports** and then selecting **New Waveguide Port**.
2. Click the **Editcrosssectionpage** tab.

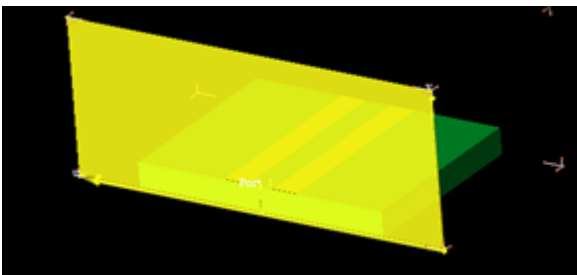
3. Type the lower limit of the port dimension in **U** and **V** fields.
4. Type the upper limit of the port dimension in **U** and **V** fields.
5. Select the **Auto-extend to simulation domain boundaries** option to automatically expand the port dimensions according to the domain boundary.

## Using the Auto-extend Option

If the Auto-extend option is not enabled, a face is selected by default:



If the Auto-extend option is enabled, the port dimensions extend automatically according to the domain boundary:



## Specifying Port Properties

You can define a port name, number of modes, de-embedding distance, impedance definition used to re-normalize S-Parameters, and Waveguide Port Definition (nodal or modal S-Parameters).

To define port properties:

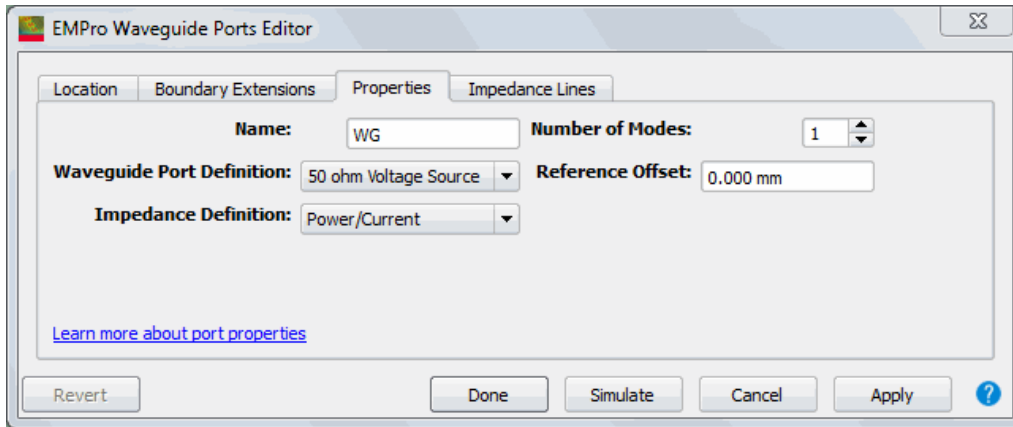
1. Open the EMPro Waveguide Ports Editor by right-clicking **Circuit Components/Ports** and then selecting **New Waveguide Port**.
2. Select a face.
3. Click the **Properties** tab in the EMPro Waveguide Ports Editor.
4. Type a name for your waveguide.
5. Select **50 ohm Voltage source** from the **Waveguide Port Definition** drop-down list. For more information, see [Waveguide Port Definition](#).
6. Specify the number of modes for this port from the Number of Modes field. For more information, see [Modes](#).
7. Select a value from the **Impedance Definition** drop-down list.
8. Specify a value in the Reference Offset field. For more information, see [Reference Offsets](#).

## Specifying Waveguide Port Definition

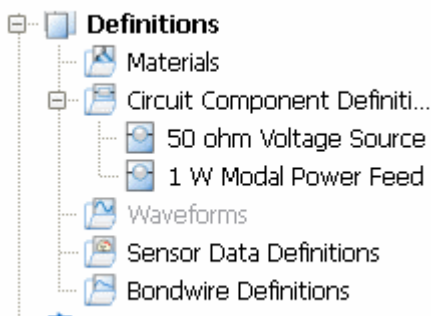
Two feed definitions are automatically created:

- 50 ohm Voltage Source results in 50 Ohm re-normalized S-Parameters.
- 1 W Modal Power Feed results in generalized S-Parameters.

A waveguide port supports Voltage source and 1 W modal power feed.



When waveguide port is chosen automatically, both Voltage source and 1 W Modal power feed is added in circuit component definition.



If you choose 1 W Modal power feed and define two modes for one waveguide port, the reference impedance for this mixed mode results are  $Z_{\text{odd}}$  and  $Z_{\text{even}}$ . For two modes port, the modes are odd and even. You can determine the corresponding mode of a port by checking its E-field distribution. If the E-field distribution of port 1 is symmetrical, it is an even mode. If the E-field distribution is unsymmetrical, it is an odd mode.

### NOTE

If a waveguide port is using 50 ohm feed, the result is single end mode.

If there is a voltage source already present, Waveguide ports will only add 1 W modal power source.

To set up a waveguide port with voltage source, refer to [Create and Simulate a Microstrip Line](#).

To set up a waveguide port with 1 W modal power source, refer to [Performing Multimode Analysis on Rectangular Waveguide](#). Modal power feed is useful if modal analysis of the structure needs to be carried.

## Nodal vs Modal Ports

Modal S-parameters to Nodal (or Single Ended) S-parameter conversion requires defined characteristic impedance at ports

The port's characteristic impedance is computed in three ways:

- Z<sub>pi</sub>: Power-current relationship, well defined and Momentum also uses it

$$Z_{pi} = \frac{2P}{I \bullet I}$$

- Z<sub>p<sub>v</sub></sub>: Power-voltage relationship, requires the impedance line to calculate the voltage

$$Z_{pv} = \frac{V \bullet V}{2P}$$

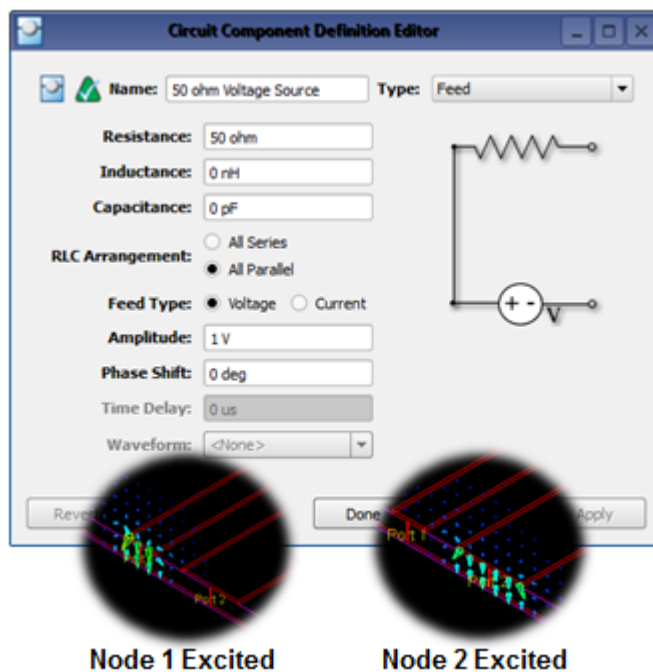
- Z<sub>vi</sub>: Voltage-current relationship, requires the impedance line to calculate the voltage

$$Z_{vi} = \sqrt{Z_{pi} Z_{pv}}$$

In TEM, the impedance become the same.

Nodal Ports provides the following features:

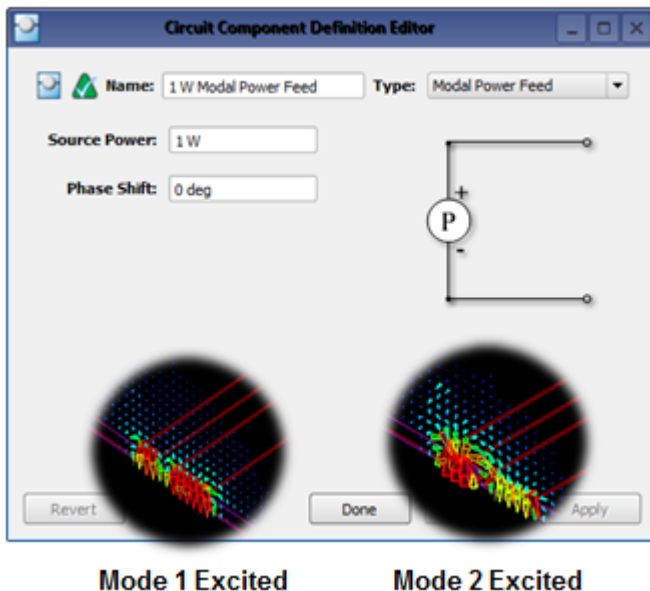
- Generate traditional S-parameters.
- Each row/column of S-matrix associated with a pin on a trace.
- Generated in EMPro when "Circuit Component "Type" is "Feed".
- Always generated for FEM in ADS.
- Recommended when transferring to circuit simulation.



Modal Ports:

Modal Ports provide the following features:

- Generates Generalized" S-parameters.
- Each row/column of S-matrix associated with a waveguide mode.
- Generated in EMPro when "Circuit Component "Type" is "Modal Power Feed".
- Analogous to differential/common S-parameters.
- Useful when analyzing transitions between different impedances, or for strongly frequency dependent line impedance.



## Number of Modes

You should specify the number of modes to include all the propagating modes in the port for a specified frequency. Each mode results in creating a set of S-parameters. Therefore, if you are analyzing two modes at each port in a three-port structure, the final result is a 6x6 S-matrix. In general, an n-port solution is the total number of excitations of all ports, the number of modes, plus the number of sources.

Each higher-order mode represents a different field pattern that can propagate down a waveguide. In general, all propagating modes should be included in a simulation. You can accept the default of 1 mode, but where propagating higher-order modes are present you need to change this to include higher-order modes. If there are more propagating modes than the number specified, a warning will be displayed when you attempt to solve the structure. You can increase the number of modes, then simulate. The number of modes can vary among ports.

Note that choosing more modes returns more detail but it does not increase the simulation time proportionally, the increase in simulation time will be less than proportional. If you choose not to include some higher-order modes in a simulation, make sure the cross sections on the ports are long enough so that the modes die out and are not reflected back.

## Defining Reference Offsets

You can select the following types of reference offsets:

- Specifying a positive reference offset: Shifts the reference planes in the direction towards the circuit.
- Specifying a negative reference offset: Shifts the reference planes in the direction away from the circuit.

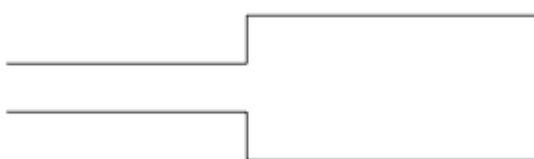


## Why Use Reference Offsets?

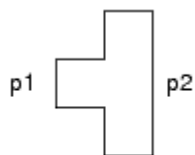
The need to adjust the position of ports in a layout is analogous to the need to eliminate the effect of probes when measuring hardware prototypes. When hardware prototypes are measured, probes are connected to the input and output leads of the Device Under Test (DUT). These probes feed energy to the DUT, and measure the response of the circuit. However, the measured response characterizes the entire setup, that is, the DUT plus the probes. This is an unwanted effect. The final measurements should reflect the characteristics of the DUT alone. The characteristics of the probes are well known, so measurement labs can mathematically eliminate the effects of the probes, and present the correct measurements of the DUT.

However, for a calibration process, it is necessary that the fundamental mode is characterized accurately. This can only be accomplished when the distance between the port boundary and the first discontinuity is sufficiently large, that is, there exists a feed line that is long enough to provide this distance.

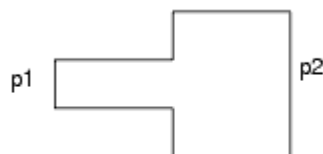
As a basic example, consider a line width that varies abruptly in some part of your circuit, as shown in the following figure:



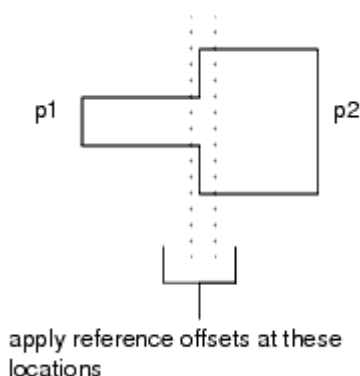
You need to characterize only the variation of the step-in-width itself, as shown in the following figure:



As mentioned previously, it takes a little distance for the fundamental mode to settle, which means that this short structure might not yield the accuracy that you expect from the simulation. In this case, allow for some feed line length, as shown in the following figure:



Now, the simulation will yield accurate results, but the results will also contain the extra line lengths. To resolved this issue, use reference offsets. Although, the circuit has been calculated with the long lines, reference offset shifting allows you to produce the S-parameters as if the short structure had been simulated instead, as shown in the following figure:

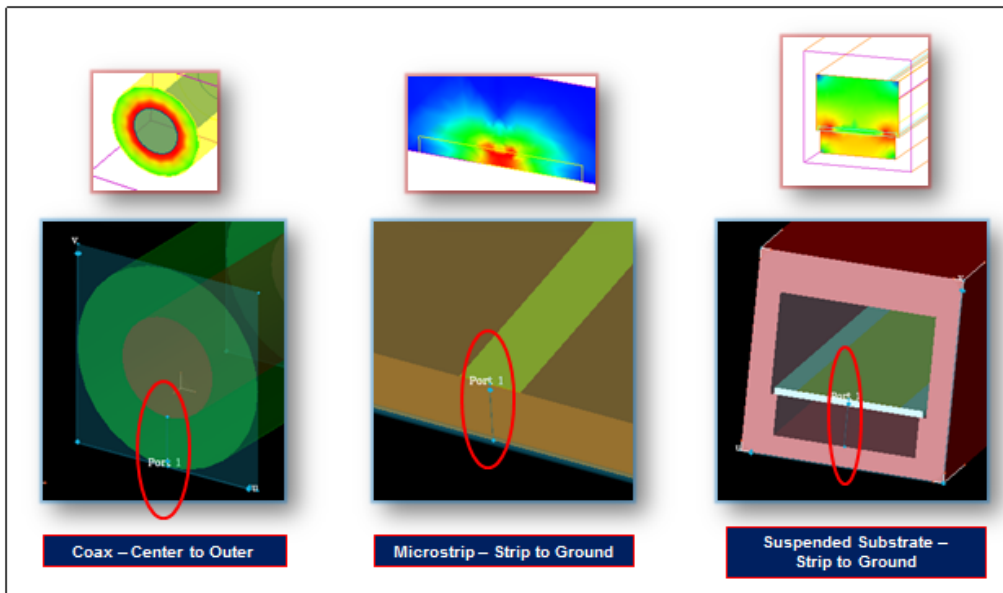


The effect of the extra feed lines is mathematically eliminated from the S-parameter solution. This process of adding or subtracting line length is generally referred to as de-embedding. During the solution process, the impedance and propagation constant has been calculated for the ports, based on their physical location in the circuit. When you know the impedance, propagation constant, and the distance of de-embedding, you can cancel out the extra lengths of line from the S-parameter results, by compensating for the loss and phase shifts of those lines. The net result is a set of S-parameters, calculated as if the extra line lengths were not there.

## Creating an Impedance Line

An impedance line is used to convert modal representation into nodal representation. An impedance line is used to compute the voltage used for Zpv and Zvi impedance computation. You need to keep the direction the same at each end (strip->ground or ground->strip) for correct transmission phase (if you flip the arrow on one end, you get a 180 degree phase error). Zpi is automatically computed.

- Z is determined by impedance line.
- Zpi (power/current) is the preferred impedance model and corresponds to Momentum.



## About Impedance Lines



The S-matrices that are initially calculated are generalized S-matrices normalized to the impedances of each port. However, you may need to compute S-matrices that are normalized to specific impedances such as 50 ohms. To convert a generalized S-matrix to a renormalized S- matrix, EMPro first computes the characteristic impedance at each port.

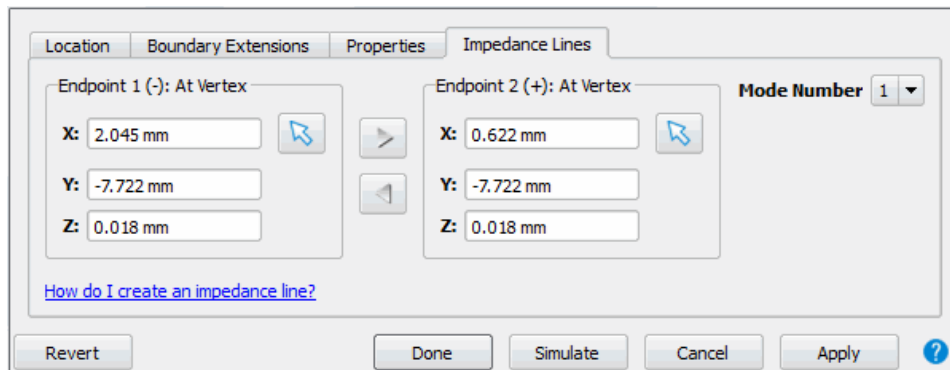
In general, pick two points at which the voltage differential is expected to be maximum. For example, on a microstrip port, place one point in the center of the microstrip, and the other directly underneath it on the ground plane. In a rectangular waveguide, place the two points in the center of the long sides.

If you are analyzing more than one mode at a port, you must define a separate set of impedance lines for each mode. The orientation of the electric field differs from mode to mode.

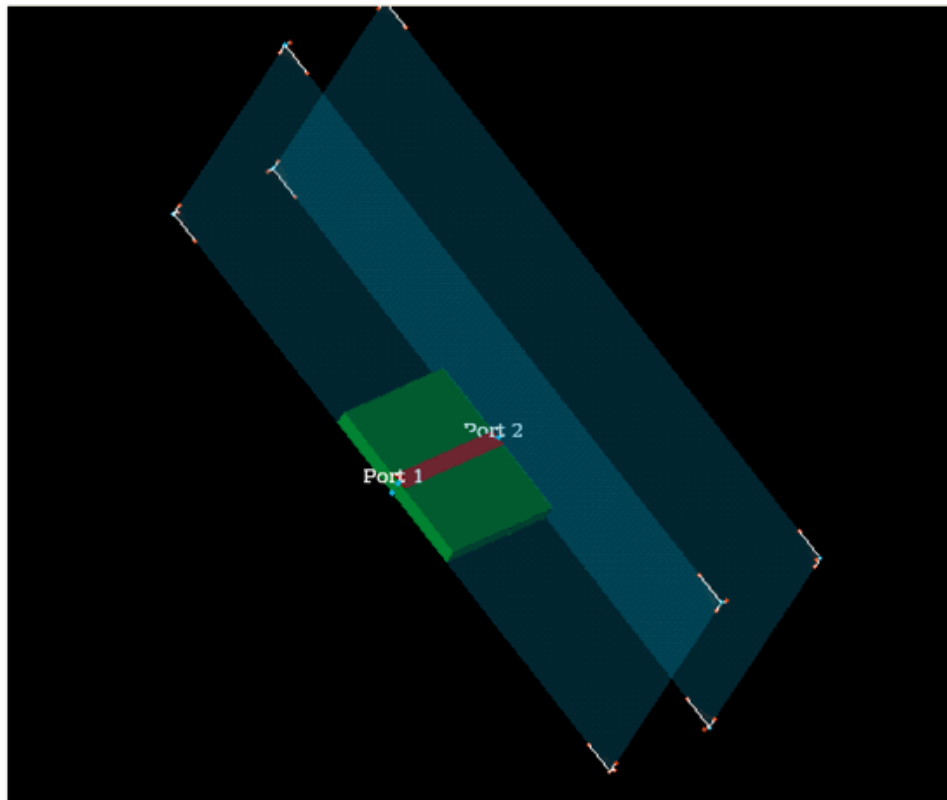
## Creating Impedance Lines

You need to define an impedance line to specify a port mode. To define an impedance line:

1. Open the EMPro Waveguide Ports Editor by right-clicking **Circuit Components/Ports** and then selecting **New Waveguide Port**.
2. Click the **Impedance Lines** tab.
3. Select the start point of the line by either typing the coordinates in the X, Y, and Z fields or by clicking  in **Endpoint1** to select the X dimensions.
4. Select the end point of the line by either typing the coordinates in the X, Y, and Z fields or by clicking  in **Endpoint2** to select the X dimensions. The endpoint2 values defines the direction and length of the impedance line.



5. Click **Done**. The impedance line appear on the waveguide port. The Waveguide port that you have defined still shows an invalid symbol. As per the message waveguide port should lie on the faces of the geometry. You need to change padding in x direction. Both lower and upper padding in X direction should be 0mm as waveguide port lie on X plane.
6. Open the FEM Padding Editor by double clicking **FEM Padding** in the project tree.
7. Select **Custom** and type **0 mm** in **Lower-X** field.
8. Click **Done**. This makes the waveguide port valid.
9. Similarly, define another waveguide port at X=7.5 mm plane. Once both waveguide ports are defined, the waveguide ports appear on the structure, as shown in the following figure:



# Calculating S-Parameters in FEM Simulation

A generalized S-matrix describes what fraction of power associated with a given field excitation is transmitted or reflected at each port.

The S-matrix for a three port structure is shown below:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

where:

- All quantities are complex numbers.
- The magnitudes of a and b are normalized to a field carrying one watt of power.
- $|a_i|^2$  represents the excitation power at port i.
- $|b_i|^2$  represents the power of the transmitted or reflected field at port i.
- The full field pattern at a port is the sum of the port's excitation field and all reflected/transmitted fields.
- The phase of  $a_i$  and  $b_i$  represent the phase of the incident and reflected/transmitted field at  $t=0$ ,  $\angle a_i$
- Represents the phase angle of the excitation field on port i at  $t=0$ . (By default, it is zero.)  $\angle b_i$
- Represents the phase angle of the reflected or transmitted field with respect to the excitation field.
- $S_{ij}$  is the S-parameter describing how much of the excitation field at port j is reflected back or transmitted to port i.

For example,  $S_{31}$  is used to compute the amount of power from the port 1 excitation field that is transmitted to port 3.

The phase of  $S_{31}$  specifies the phase shift that occurs as the field travels from port 1 to port 3.

## NOTE

When the 2D solver computes the excitation field for a given port, it has no information indicating which way is *up* or *down*. Therefore, if ports have not been calibrated, it is possible to obtain solutions in which the S-parameters are out of phase with the expected solution.

## Frequency Points

The S-parameters associated with a structure are a function of frequency. Therefore, separate field solutions and S-matrices are generated for each frequency point of interest. FEM Simulator supports two types of frequency sweeps:

- Discrete frequency sweeps, in which a solution is generated for the structure at each frequency point you specify.
- Fast frequency sweeps, in which asymptotic waveform evaluation is used to extrapolate solutions for a range of frequencies from a single solution at a center frequency.

Fast frequency sweeps are useful for analyzing the behavior of high Q structures. For wide bands of information, they are much faster than solving the problem at individual frequencies.

#### NOTE

Within a fast frequency solution, there is a bandwidth where the solution results are most accurate. This range is indicated by an error criterion using a matrix residue that measures the accuracy of the solution. For complex frequency spectra that have many peaks and valleys, a fast sweep may not be able to accurately model the entire frequency range. In this case, additional fast sweeps with different expansion frequencies will automatically be computed and combined into a single frequency response.

## Z- and Y-Matrices

Calculating and displaying the unique impedance matrices (Z) associated with a structure is performed in the post processor.

### Characteristic Impedances

FEM Simulator calculates the characteristic impedance of each port in order to compute a renormalized S-matrix, Z-matrix, or Y-matrix. The system computes the characteristic impedance of each port in three ways—as  $Z_{pi}$ ,  $Z_{pv}$ , and  $Z_{vi}$  impedances.

You have the option of specifying which impedance is to be used in the renormalization calculations.

### PI Impedance

The  $Z_{pi}$  impedance is the impedance calculated from values of power (P) and current (I):

$$Z_{pi} = \frac{2P}{I \cdot I}$$

The power and current are computed directly from the simulated fields. The power passing through a port is equal to the following:

$$P = \frac{1}{2} \oint_S \mathbf{E} \times \mathbf{H} \cdot d\mathbf{s}$$

where the surface integral is over the surface of the port.

The current is computed by applying Ampere's Law to a path around the port:

$$I = \oint_l \mathbf{H} \cdot d\mathbf{l}$$

While the net current computed in this way will be near zero, the current of interest is that flowing into the structure,  $I^-$  or that flowing out of the structure,  $I^+$ . In integrating around the port, the system keeps a running total of the contributions to each and uses the average of the two in the computation of impedances.

## PV Impedance

The  $Z_{pv}$  impedance is the impedance calculated from values of power (P) and voltage (V):

$$Z_{pv} = \frac{V \cdot V}{2P}$$

where the power and voltage are computed directly from the simulated fields. The power is computed in the same way as for the  $Z_{pi}$  impedance. The voltage is computed as follows:

$$V = \oint_l \mathbf{E} \cdot d\mathbf{l}$$

The path over which the system integrates is referred to as the impedance line, which is defined when setting up the ports. To define the impedance line for a port, select the two points across which the maximum voltage difference occurs. FEM Simulator cannot determine where the maximum voltage difference will be unless you define an impedance line.

## VI Impedance

The  $Z_{vi}$  impedance is given by:

$$Z_{vi} = \sqrt{Z_{pi} Z_{pv}}$$

For TEM waves, the  $Z_{pi}$  and  $Z_{pv}$  impedances form upper and lower boundaries to a port's actual characteristic impedance. Therefore, the value of  $Z_{vi}$  approaches a port's actual impedance for TEM waves.

## Choice of Impedance

- When the system is instructed to renormalize the generalized S-matrix or compute a Y- or Z-matrix, you must specify which value to use in the computations,  $Z_{pi}$ ,  $Z_{pv}$ , or  $Z_{vi}$ .
- For TEM waves, the  $Z_{vi}$  impedance converges on the port's actual impedance and should be used.
- When modeling microstrips, it is sometimes more appropriate to use the  $Z_{pi}$  impedance.
- For slot-type structures (such as finline or coplanar waveguides),  $Z_{pv}$  impedance is the most appropriate.

## De-embedding

If a uniform length of transmission line is added to (or removed from) a port, the S-matrix of the modified structure can be calculated using the following relationship:

$$[S'] = [e^{\gamma l}] [S] [e^{\gamma l}]$$

Where,  $\epsilon^{\gamma}$  is a diagonal matrix with the following entries:

$$\begin{bmatrix} e^{\gamma_1 l_1} & 0 & 0 \\ 0 & e^{\gamma_2 l_2} & 0 \\ 0 & 0 & e^{\gamma_3 l_3} \end{bmatrix}$$

$\gamma = \alpha + j\beta$  is the complex propagation constant, where:

- $\alpha$  is the attenuation constant of the wave of port  $i$ .
- $\beta$  is the propagation constant associated with the uniform transmission line at port  $i$ .
- $l_i$  is the length of the uniform transmission line that has been added to or removed from the structure at port  $i$ . A positive value indicates that a length of transmission line has been removed from the structure.

The value of  $\beta$  for the dominant mode of each port is automatically calculated by the 2D solver.

## Equations

The sections below describe some of the equations that are solved in a simulation or used to define elements of a structure.

## Derivation of Wave Equation

The solution to the following wave equation is found during a simulation:

$$\nabla \times \left( \frac{1}{\mu_r} \nabla \times \mathbf{E} \right) - k_0^2 \epsilon_r \mathbf{E} = 0$$

where:

- $\mathbf{E}(x,y,z)$  is a phasor representing an oscillating electric field
- $k_0$  is the free space wave number,  $\omega \sqrt{\mu_0 \epsilon_0}$ .
- $\omega$  is the angular frequency,  $2\pi f$ .
- $\mu_r(x,y,z)$  is the complex relative permeability.
- $\epsilon_r(x,y,z)$  is the complex relative permittivity.

The difference between the 2D and 3D solvers is that the 2D solver assumes that the electric field is a traveling wave with this form:

$$\mathbf{E}(x,y,z,t) = \text{Re}[\mathbf{E}(x,y) e^{(j\omega t - \gamma z)}]$$

while the 3D solver assumes that the phasor  $\mathbf{E}$  is a function of  $x$ ,  $y$ , and  $z$ :



$$\mathbf{E}(x,y,z,t) = \Re[\mathbf{E}(x,y,z)e^{j\omega t}]$$

## Maxwell Equations

The field equation solved during a simulation is derived from Maxwell's Equations, which in their time-domain form are:

$$\nabla \times \mathbf{H}(t) = \mathbf{J}(t) + \frac{\partial}{\partial t} \mathbf{D}(t)$$

$$\nabla \times \mathbf{E}(t) = -\frac{\partial}{\partial t} \mathbf{B}(t)$$

$$\nabla \cdot \mathbf{D}(t) = \rho$$

$$\nabla \cdot \mathbf{B}(t) = 0$$

Where:

- $\mathbf{E}(t)$  is the electric field intensity.
- $\mathbf{D}(t)$  is the electric flux density,  $\mathbf{E}(t)$ , and  $\epsilon$  is the complex permittivity.
- $\mathbf{H}(t)$  is the magnetic field intensity.
- $\mathbf{B}(t)$  is the magnetic flux density,  $\mu \mathbf{H}(t)$ , and  $\mu$  is the complex permeability.
- $\mathbf{J}(t)$  is the current density,  $\mathbf{E}(t)$ .
- $\rho$  is the charge density.

## Phasor Notation

Because all time-varying electromagnetic quantities are oscillating at the same frequency, they can be treated as phasors multiplied by  $e^{j\omega t}$  (in the 3D solver) or by  $e^{j\omega t - z}$  (in the 2D solver).

In the general case with the 3D solver, the equations become:

$$\nabla \times \mathbf{H}e^{j\omega t} = \mathbf{J}e^{j\omega t} + \frac{\partial}{\partial t} \mathbf{D}e^{j\omega t}$$

$$\nabla \times \mathbf{E}e^{j\omega t} = -\frac{\partial}{\partial t} \mathbf{B}e^{j\omega t}$$

$$\nabla \cdot \mathbf{D}e^{j\omega t} = \rho e^{j\omega t}$$

$$\nabla \cdot \mathbf{B}e^{j\omega t} = 0$$

By factoring out the quantity  $e^{j\omega t}$  and using the following relationships:

$$\frac{\partial}{\partial t} \mathbf{E}e^{j\omega t} = j\omega \mathbf{E}e^{j\omega t}$$

$$\frac{\partial \mathbf{H}}{\partial t} = j\omega \mathbf{H}$$

Maxwell's Equations in phasor form reduce to:

$$\nabla \times \mathbf{H} = \mathbf{J} + j\omega \mathbf{E}$$

$$\nabla \times \mathbf{E} = -j\omega \mathbf{B}$$

$$\nabla \cdot \mathbf{D} = \rho$$

$$\nabla \cdot \mathbf{B} = 0$$

where  $\mathbf{B}$ ,  $\mathbf{H}$ ,  $\mathbf{E}$ , and  $\mathbf{D}$  are phasors in the frequency domain. Now, using the relationships  $\mathbf{B} = \mu \mathbf{H}$ ,  $\mathbf{D} = \epsilon \mathbf{E}$ , and  $\mathbf{J} = \sigma \mathbf{E}$ , Maxwell's Equations in phasor form become:

$$\nabla \times \mathbf{H} = (j\omega\epsilon + \sigma) \mathbf{E} = j(\omega\epsilon) \mathbf{E}$$

for  $\sigma = 0$

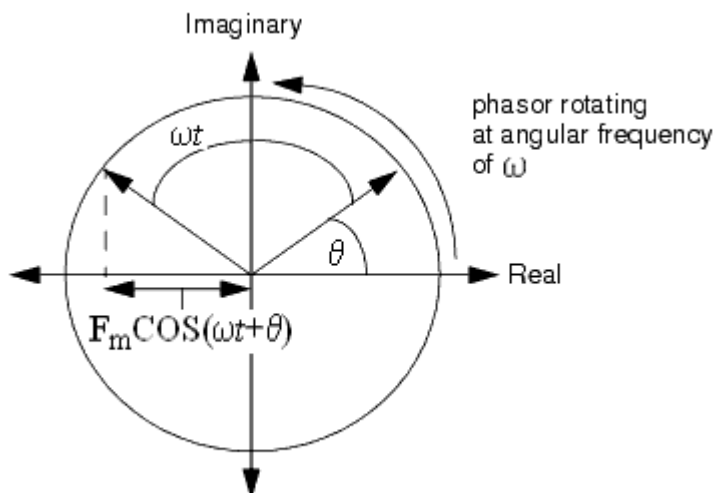
$$\nabla \times \mathbf{H} = j(\omega\epsilon) \mathbf{E}$$

$$\nabla \cdot \epsilon \mathbf{E} = \rho$$

$$\nabla \cdot \mu \mathbf{H} = 0$$

Where:

- $\mathbf{H}$  and  $\mathbf{E}$  are phasors in the frequency domain,  $\mu$  is the complex permeability, and  $\epsilon$  is the complex permittivity.
- $\mathbf{H}$  and  $\mathbf{E}$  are stored as phasors, can be visualized as a magnitude and phase or as a complex quantity.



## Electric and Magnetic Fields Can Be Represented as Phasors

### Assumptions

To generate the final field equation, place  $\mathbf{H}$  in the equation in terms of  $\mathbf{E}$  to obtain:

$$\mathbf{H} = \frac{1}{j\omega\mu} \nabla \times \mathbf{E}$$

Then, substitute this expression for  $\mathbf{H}$  in the  $\nabla \times \mathbf{H}$  equation to produce:

$$\nabla \times \left( -\frac{1}{j\omega\mu} \nabla \times \mathbf{E} \right) = j\omega\epsilon \mathbf{E}$$

### Conductivity

Although good conductors can be included in a model, the system does not solve for any fields inside these materials. Because fields penetrate lossy conductors only to one skin depth (which is a very small distance in good conductors), the behavior of a field can be represented with an equivalent impedance boundary.

For perfect conductors, the skin depth is zero and no fields exist inside the conductor. Perfect conductors are assumed to be surrounded with Perfect E boundaries.

### Dielectric Loss Tangent

Dielectric losses can be modeled by assuming that the relative permittivity,  $\hat{\epsilon}_r$ , is complex:

$$\hat{\epsilon} = \epsilon' - \epsilon''$$

Expressed in terms of the dielectric (electric) loss tangent,  $\tan \delta_e = \epsilon''_r / (\epsilon'_r)$ , the complex relative permittivity,  $\hat{\epsilon}_r$  becomes:

$$\hat{\epsilon}_r = \epsilon'_r - j\epsilon'_r \tan \delta_e$$

### Magnetic Loss Tangent

Losses in magnetic materials can be modeled by assuming that  $\mu_r$  is complex.

$$\hat{\mu}_r = \mu'_r - j\mu''_r$$

Expressed in terms of the magnetic loss tangent,  $(\mu''_r)/(\mu'_r)$ , the complex relative permeability becomes:

$$\hat{\mu}_r = \mu_r' \left( 1 - \frac{j\mu_r''}{\mu_r'} \right) = \mu_r' (1 - j \tan \delta_m)$$

## Definition of Freespace Phase Constant

Using the relationships  $\epsilon = \epsilon_0 \epsilon_r$  and  $\mu = \mu_0 \mu_r$ , the wave equation being solved can be placed in this form:

$$\nabla \times \left( \frac{1}{\mu_r} \nabla \times \mathbf{E} \right) - \omega^2 \mu_0 \epsilon_0 \epsilon_r \mathbf{E} = 0$$

Now, if the freespace phase constant (or wave number) is defined as,  $k_0^2 = \omega^2 \mu_0 \epsilon_0$ , the above reduces to:

$$\nabla \times \left( \frac{1}{\mu_r} \nabla \times \mathbf{E} \right) - k_0^2 \epsilon_r \mathbf{E} = 0$$

which is the equation that the 2D and 3D engines solve.

# Using FEM RLC Loads

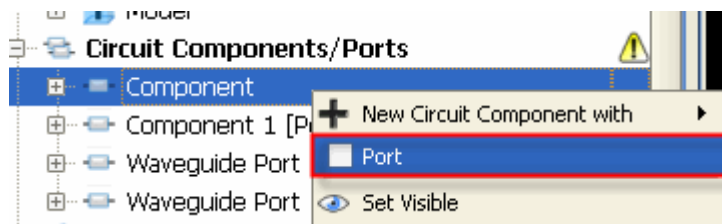
You can include RLC loads directly in an FEM simulation for representing matching circuits, SMD components, and other passive loads. Previously, you had to export s-parameters to a circuit simulator and then set up circuit simulations, which was time consuming and generated errors especially for designs with multiple ports. Also, only S-parameter results were displayed and passive networks were not included in field visualization or radiation patterns.

Now, you can define passive loads directly in the FEM setup environment, which consists of common series and parallel RLC network topologies. The following features are supported in FEM:

- You can define an RLC discrete circuit component as a passive load in Circuit Component Definitions. You can assign this passive load to a component attached to the geometry. The circuit system and FEM system are solved together only once.
- Instead of an open system, a loaded system is solved for FEM. It provides a better matrix condition and faster converging (around 1.4x – 3x) for iterative matrix solver with circuit sources.

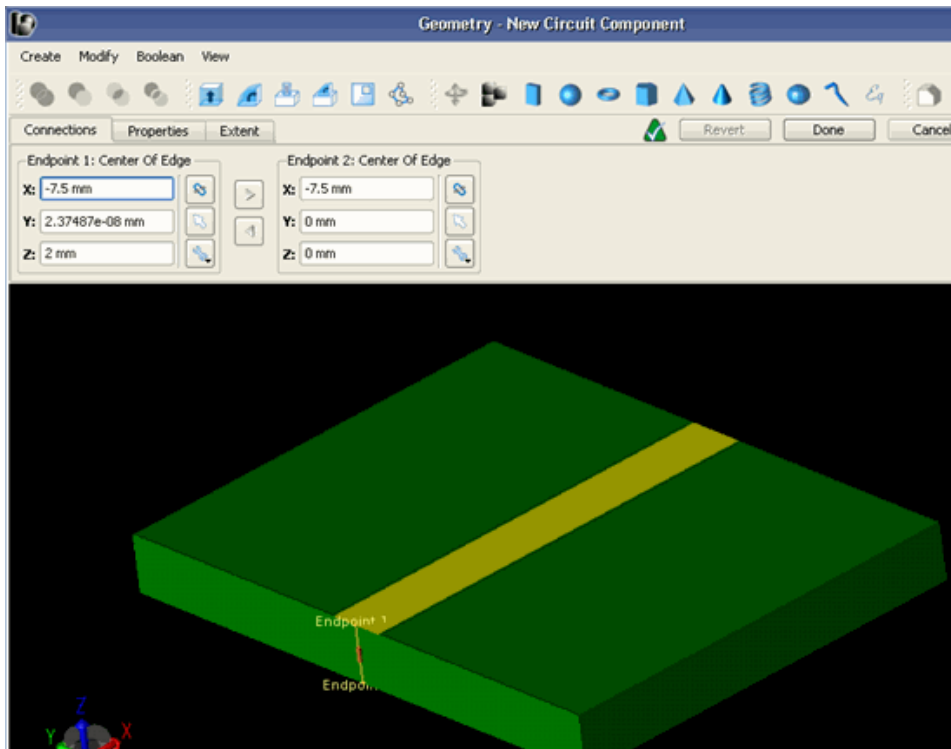
## Adding a Passive Load Component

Before adding a component, remove selection from the Port check box, as shown in the following figure:



To add a passive load component in an FEM simulation:

1. Select **FEM Simulation** from the drop-down list on the **Simulation** toolbar.
2. Right-click **Circuit Components/Ports** > **New Circuit Component with** > **New Passive Load Definition**. The New Circuit Component window is displayed in the Geometry workspace.
3. Specify the coordinates for endpoint 1 and endpoint 2 in the **Connections** tab, as shown in the following figure:

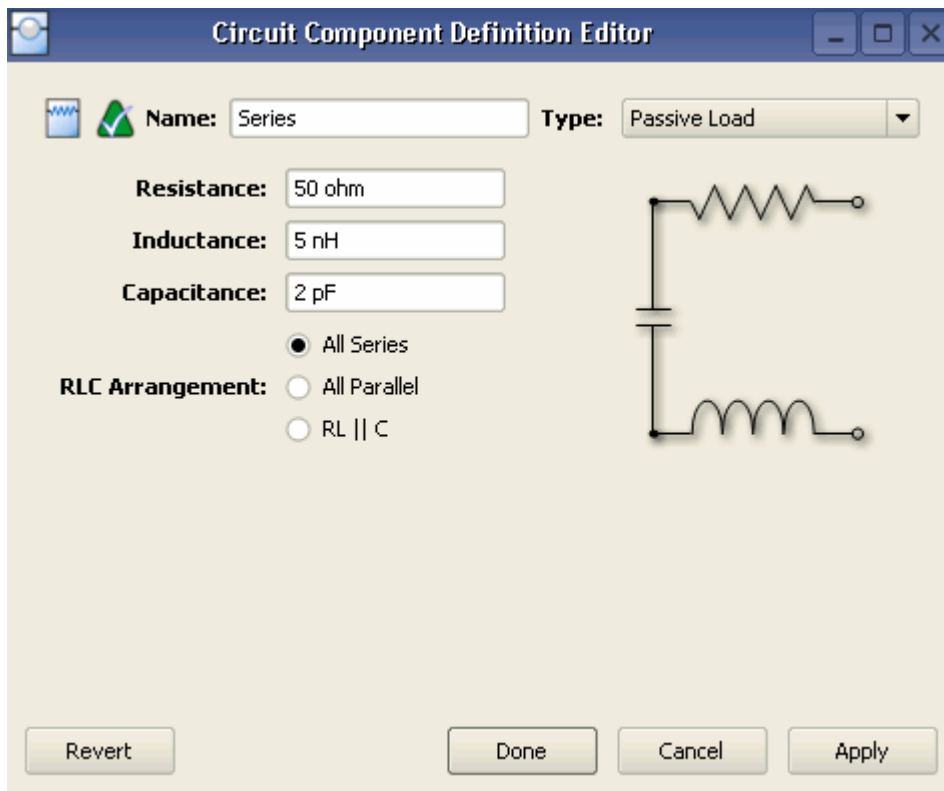


4. Click the **Properties** tab.
5. Type a component name in the **Name** text box.
6. Click **Done**.

## Adding a Passive Load Definition

To add a passive load circuit component definition:

1. Select **FEM Simulation** from the drop-down list on the **Simulation** toolbar.
2. Right-click **Circuit Components Definitions** > **New Circuit Component Definition**. The Circuit Component Definition Editor is displayed, as shown in the following figure:



3. Type a component name in the **Name** text box.
4. Select **Passive load** from the **Type** drop-down list.
5. Type a value in the **Resistance** text box.
6. Type a value in the **Inductance** text box.
7. Type a value in the **Capacitance** text box.
8. Select one of the following options in **RLC Arrangement**:
  - All Series
  - All Parallel

**NOTE**

FEM does not support the **RL || C** option.

9. Click **Apply**.
10. Click **Done**.

## Adding a Feed Definition

To add a feed circuit component definition:

1. Select **FEM Simulation** from the drop-down list on the **Simulation** toolbar.
2. Right-click **Circuit Components Definitions** > **New Circuit Component Definition**. The Circuit Component Definition Editor dialog box is displayed.
3. Type a component name in the **Name** text box.

- 
4. Select **Feed** from the **Type** drop-down list.
  5. Type a value in the **Resistance** text box.

**NOTE**

Since the RLC values in the Feed setup are also used to compute the reference impedance of this port, always use real numbers for the reference impedance. For a Feed setup, assign 0 values for **Inductance** and **Capacitance** options.

6. Select one of the following options in **RLC Arrangement**:
  - All Series
  - All Parallel
7. Select a feed type option.
8. Type a value in the **Amplitude** text box.
9. Type a value in the **Phase Shift** text box.
10. Type a value in the **Time Delay** text box.
11. Select an option from the Waveform drop-down list.
12. Click **Apply**.
13. Click **Done**.

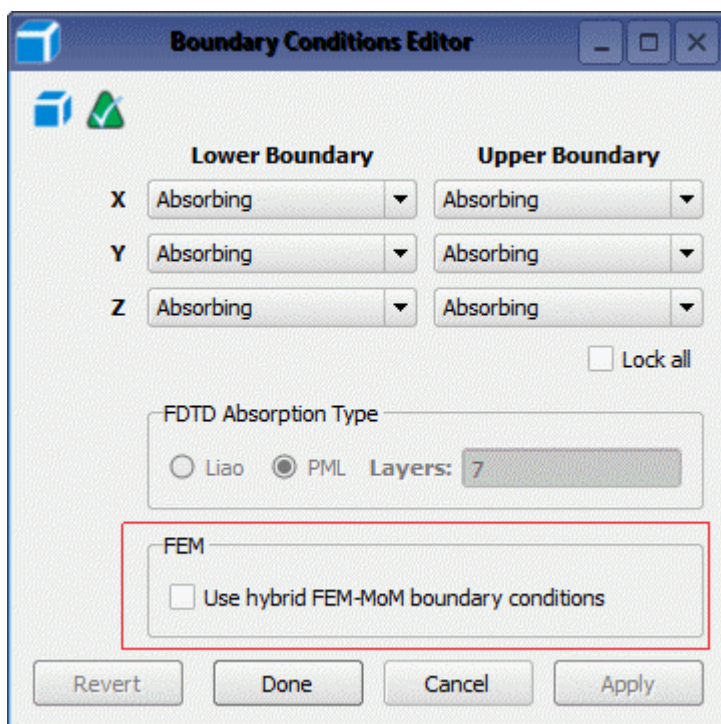


# Hybrid FEM-MoM Boundary Conditions

In FEM simulations, the accuracy of results in the space outside a bounding box is limited by the accuracy of boundary conditions. These boundary conditions are based on simplifying approximations, which often require a large padding. The approximations can be avoided by modeling the outside space using Method of Moments (MoM). You can integrate the flexibility and efficiency of FEM inside the bounding box with the superior accuracy of MoM in the open space outside a bounding box. The FEM-MoM hybrid simulation can improve result accuracy, if the electromagnetic fields outside a bounding box are important, such as for antennas, inductors, and waveguides. However, it is not supported for a combination of waveguide port simulations.


To specify MoM hybrid boundary conditions for radiating problems:

1. Double-click **Boundary Conditions** in the Simulation Domain branch of the project tree. The Boundary Condition Editor is displayed.



2. Specify the upper and lower boundary conditions.
3. Select the **Use hybrid FEM-MoM boundary conditions** option.
4. Click **Done** to apply the boundary settings.

The accuracy of MoM boundary conditions depends only on the mesh size at the boundaries (which is controlled by the background mesh size). The accuracy is independent of the padding. It is recommended to select the Automatic option, as shown in the following figure:

 **Mesh/Refinement Properties - (Delta Error: 0.01)**

Stop criterium   Refinement   Initial mesh   Advanced

Background Mesh

☒ Automatic (recommended)

☐ Minimal memory

☐ Custom target mesh size:

# Examples How to Simulate using FEM

---

You can use the FEM solver to simulate various types of components such as Microstrip Line, Microstrip Line with a symmetric plane, Microstrip Line with sheet port, and Multimode Analysis on a rectangular waveguide.

This section provides information about the following topics:

- [Create and Simulate a Microstrip Line](#)
- [Simulating a Microstrip Line with Symmetric Plane](#)
- [Simulating a Microstrip Line with Sheet Port](#)
- [Performing Multimode Analysis on Rectangular Waveguide](#)
- [Performing an Eigenmode Simulation for a Rectangular Cavity](#)

## Create and Simulate a Microstrip Line

---

In this example, a Microstrip Line design is created for an FEM Simulation. This example illustrates the design of a 50 ohm microstrip line using EMPro with both FEM and FDTD simulators. The 50 ohm microstrip line is designed using substrate of dielectric constant 9.9 and thickness of 2mm.

Perform the following tasks to create a Microstrip Line design:

- Create a Microstrip Line geometry by creating a substrate and place a microstrip line on top of the substrate.
- Create and assign materials to the geometry.
- Define the outer boundary
- Assign a Waveguide port.

### Create an EMPro Project

To create a Microstrip Line geometry, create a new project by selecting **File > New Project**. Save this new project by selecting **File > Save Project**. Specify a project name.


### Creating the Microstrip line Geometry

You can create a Microstrip line geometry using a rectangular substrate and Microstrip line in the Geometry window. Using the Geometry Tools interface, you can create a rectangular substrate and rectangular Sheet Bodies.

### Creating a Substrate

You need to create a rectangular substrate named Substrate, with orientation specified from (-7.5, -10) to (7.5, 10) and 2mm extrude distance in the +Z direction.

1. Click **Extrude** on the **Create Geometry** toolbar. You can also select **Create > Geometry > Extrude**. The Create Extrude window is displayed.

2. Select the **Rectangle** tool (  ) from the **Shapes** toolbar.
3. Press the **Tab** key. The **Specify Position** dialog box is displayed. Specify the following coordinates:
  - In the **U** field, type **-7.5 mm**.
  - In the **V** field, type **-10 mm**.
4. Click **OK**.
5. Press the **Tab** key again to specify coordinates of the bottom-left vertex. The **Specify Position** dialog box is displayed. Specify the following coordinates:
  - In the **U** field, type **7.5 mm**.
  - In the **V** field, type **10 mm**.

**NOTE**



If you are unable to open the Specify Position dialog box, click once in the Geometry window and then press the Tab key.

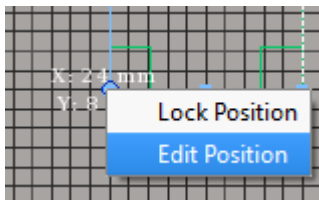
6. Accept the default width and height. A rectangle is created in the Geometry window:



7. Type **Substrate** in the **Name** text box.
8. Click the **Extrude** tab to extrude the rectangular region.
9. Type **2 mm** in the **Extrude Distance** field.
10. Click **Done** to finish the Substrate geometry.

Alternatively, you can also create a substrate by performing the following steps:

1. Click **Extrude** on the **Create Geometry** toolbar.
2. Select the **Rectangle** tool (  ) from the **Shapes** toolbar.
3. Draw a rectangle and click once in the Geometry window.
4. Click the **Select/Manipulate** (  ) to select a vertex point.
5. Right-click the top-left vertex point of the rectangle and select **Edit Position** to modify the rectangle coordinates.




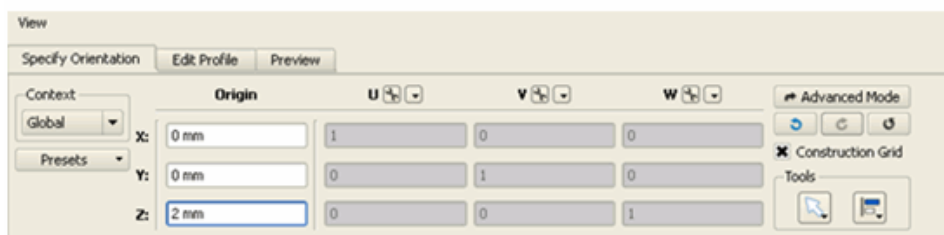
6. The **Specify Position** dialog box is displayed. Specify the following coordinates:
  - U: Type **-7.5 mm**.
  - V: Type **-10 mm**.
7. Click **OK**.
8. Right-click the bottom-right vertex point of the rectangle and select **Edit Position** to modify the rectangle coordinates.
9. The **Specify Position** dialog box is displayed. Specify the following coordinates:
  - U: Type **7.5 mm**.
  - V: Type **10 mm**.
10. Click **OK**.


## Creating a Microstrip Line

The Microstrip Line is created with a Sheet Body object that exists on the top of the Substrate. This shape consists of one rectangle. The line stretches from (-7.5, 1) to (7.5, -1).

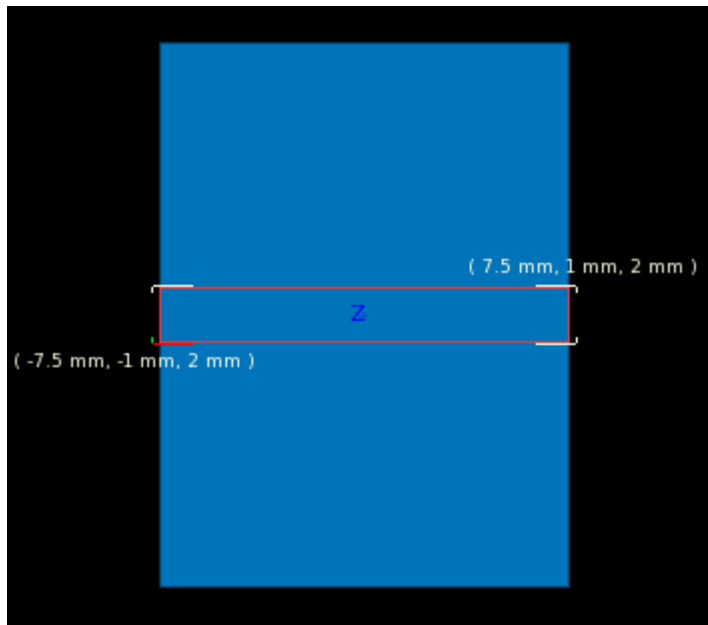
To create a microstrip line:

1. Click **Sheet Body** (  ) on the **Create Geometry** toolbar. You can also right-click the **Parts** branch of the project tree and choose **Create New > Sheet Body**. The \*Create Sheet Body window is displayed.
2. Click the **Specify Orientation** tab. Set the origin to (0, 0, 2 mm) to place the Sheet Body on top of the Substrate.






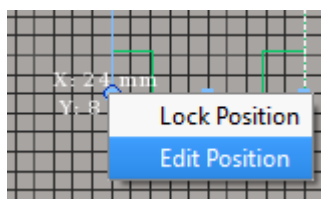
3. Click the **Edit Profile** tab. Type **Microstrip Line** in the **Name** text box.
4. Select the **Rectangle** tool (  ) from the **Shapes** toolbar.
5. Press the Tab key. The **Specify Position** dialog box is displayed. Specify the following coordinates:
  - U: Type **-7.5 mm**.
  - V: Type **-1 mm**.
6. Press the Tab key again to specify coordinates for the bottom-left vertex. The **Specify Position** dialog box is displayed. Specify the following coordinates:

- U: Type **7.5 mm**.
  - V: Type **1 mm**.
7. Click the **Extrude** tab to extrude the rectangular region.
  8. Enter a distance of 2 mm.
  9. Click **Done**. The Microstrip line is placed on the top of the substrate:



Alternatively, you can create a microstrip line by performing the following steps:

1. Click **Sheet Body** (  ) on the **Create Geometry** toolbar.
2. Select the **Rectangle** tool (  ) from the **Shapes** toolbar.
3. Draw a rectangle and click once in the Geometry window.
4. Click the **Select/Manipulate** (  ) to select a vertex point.
5. Right-click the top-left vertex point of the rectangle and select **Edit Position** to modify the rectangle coordinates.



6. The **Specify Position** dialog box is displayed. Specify the following coordinates:
  - U: Type **-7.5 mm**.
  - V: Type **-1 mm**.
7. Click **OK**.
8. Right-click the bottom-right vertex point of the rectangle and select **Edit Position** to modify the rectangle coordinates.
9. The **Specify Position** dialog box is displayed. Specify the following coordinates:

- U: Type **7.5 mm**.
- V: Type **1 mm**.

10. Click **OK**.

## Defining Mesh Priority

For an accurate calculation, ensure that the meshing priority of the Microstrip Line is greater than the Substrate. To set mesh priority:

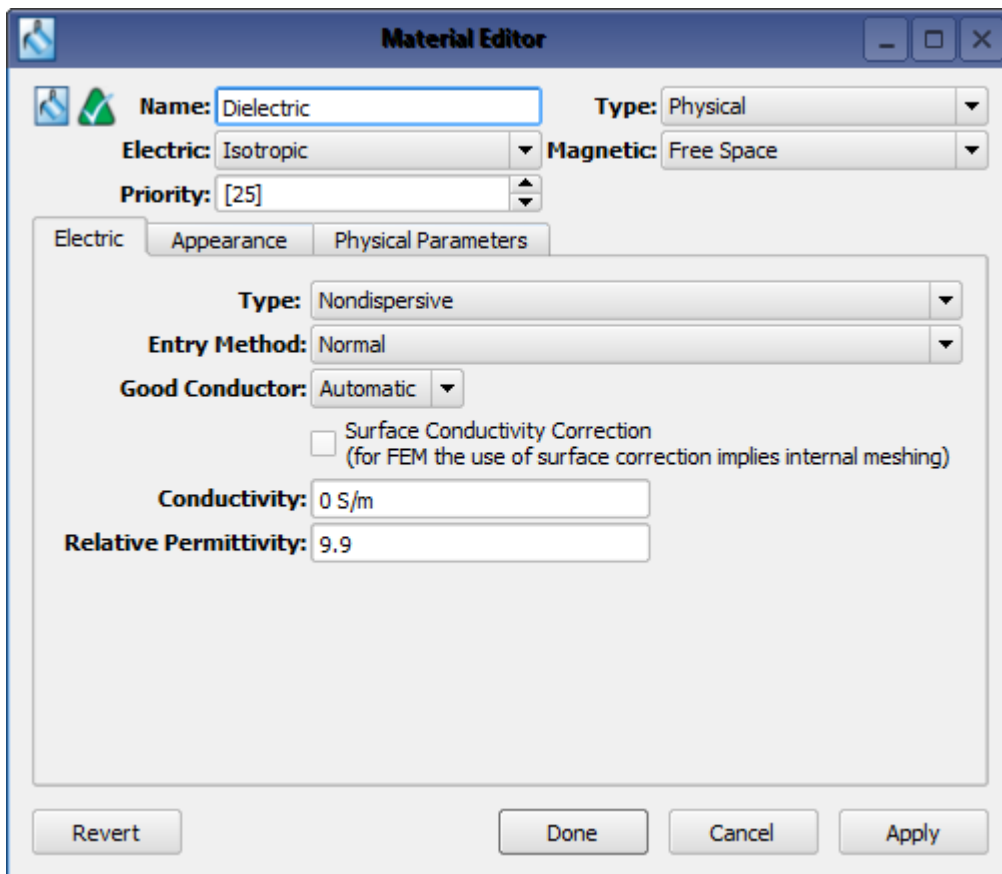
1. Right-click **Microstrip** in the project tree.
2. Select **Order > Move to Top**. The microstrip line component will be meshed before the substrate.

## Creating and Assigning Materials

You can assign a material to the Microstrip line geometry. To create a material:

1. From the **Definitions** branch of the project tree, right-click **Materials** and select **New Material Definition**. A new material object is added under Materials.
2. Double-click the new material to edit its properties. Specify the following properties for the electric conductor material:
  - **Name**: Dielectric
  - **Electric**: Isotropic
  - **Magnetic**: Freespace
  - **Relative Permittivity** = 9.9
  - **Conductivity** = 0 S/m

Accept the other default values:



Depending on your requirements, you can also set the display color of the PEC material in the **Appearance** tab.

You can also add a default material available in the library. To add **Cu** from the material library:

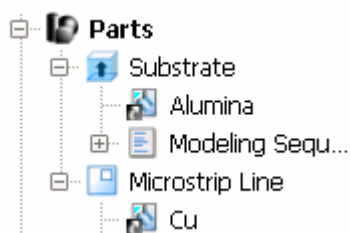
1. Right-click **Materials** in the project tree and choose the **Select from Default Material Library** option.
2. Select the material **Cu (copper)** and click **Add**. This material is added in the material list.

## Assigning Materials

To assign materials:

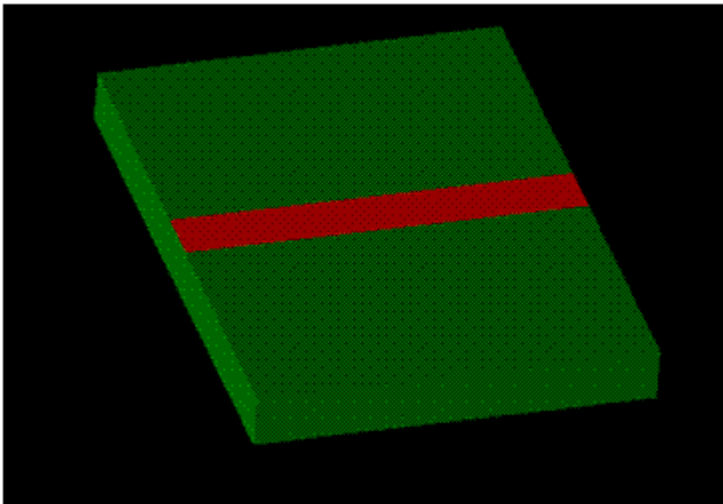
1. Click and drag the **CU** material object located in the project tree and drop it on top of the Microstrip Line objects in the **Parts** branch of the tree.
2. Assign the **Alumina** material to the Substrate object using the same procedure.

The following figure displays the project tree after material objects have been dropped on their respective parts:



The following image shows the Microstrip Line geometry with materials applied and colors set for each.





## Defining the Outer Boundary

You can define an outer boundary to specify the field behavior at the edges of the microstrip line geometry to define a boundary:

1. Double-click **Boundary Conditions** in the **Simulation Domain** branch of the project tree to open the **Boundary Conditions Editor**.
2. Set the outer boundary properties as \* Boundary: Select **Absorbing** for all boundaries except Lower Boundary Z, which should be **PEC**.
3. Click **Done** to apply the outer boundary settings.
4. Double-click **FEM Padding** in the **Simulation Domain** branch of the project tree. Set the **Upper** and **Lower** limits as shown in the following figure:
5. Click **Done**.

## Adding a Waveguide Port

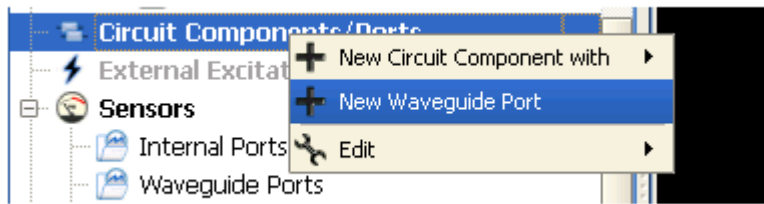
You can now add a Waveguide port to the Microstrip Line geometry. It consists of a voltage source and series 50 resistor connected between Microstrip Line and the ground plane.


A waveguide port represents the surface through which a signal can enter and exist a geometry. Therefore, you need to define two ports. A solution is generated by exciting each waveguide port individually.

## Adding a Port

Adding a port, includes selecting the object face, defining voltage, and defining an impedance line. You can perform all these tasks by using the EMPro Waveguide Ports Editor. To select a face where you want to add a port:

1. Right-click **Circuit Components/Ports** and select **New Waveguide Port**, as shown below:



2. The **EMPro Waveguide Ports Editor** is displayed. In the **Location** tab, click  to select the face of Substrate coinciding  $X=-7.5$  mm plane where you want to assign a port.



## Define Voltage

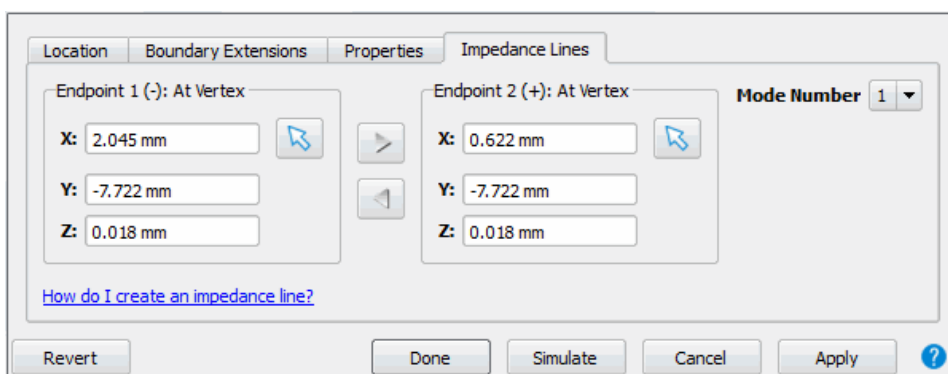
To define voltage:

1. Click the **Properties** tab in the EMPro Waveguide Ports Editor.
2. Select **50 ohm Voltage source** from the **Waveguide Port Definition** drop-down list.

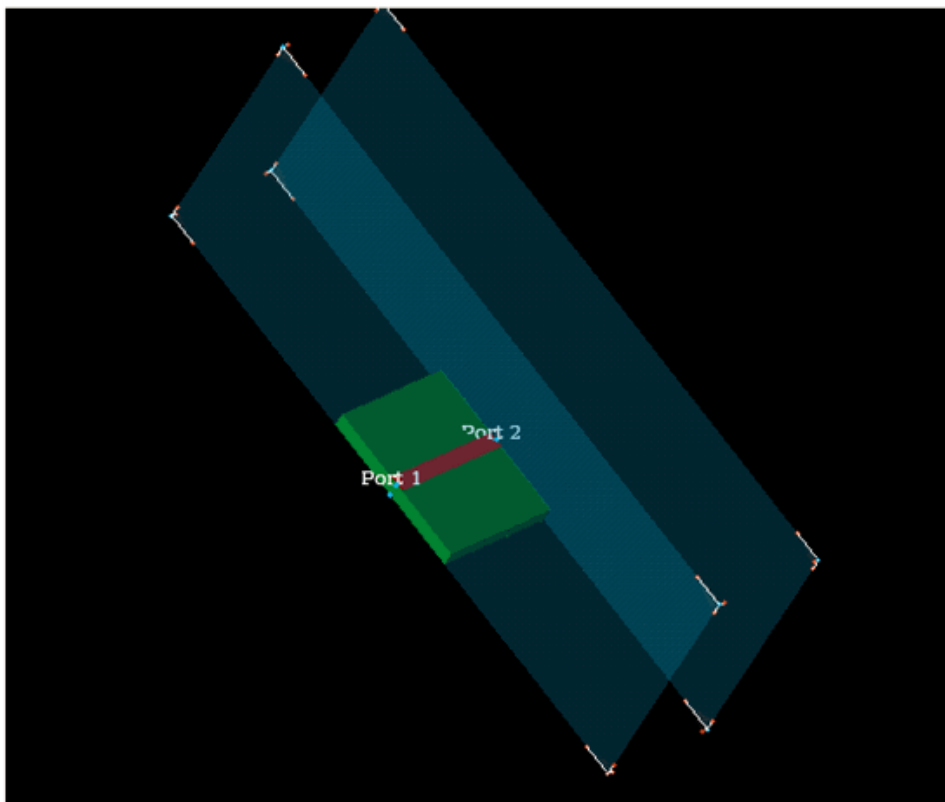
## Define Impedance line

You need to define an impedance line to specify a port mode.

1. Click the **Impedance Lines** tab.
2. Select the start point of the line by either typing the coordinates in the X, Y, and Z fields or by clicking  in **Endpoint1** to select the X dimensions.
3. Select the end point of the line by either typing the coordinates in the X, Y, and Z fields or by clicking  in **Endpoint2** to select the X dimensions. The endpoint2 values defines the direction and length of the impedance line.





4. Click **Done**. The impedance line appear on the waveguide port. The Waveguide port that you have defined still shows an invalid symbol. As per the message waveguide port should lie on the faces of the geometry. You need to change padding in x direction. Both lower and upper padding in X direction should be 0mm as waveguide port lie on X plane.
5. Click **Done**. This makes the waveguide port valid.  
Similarly, define another waveguide port at  $X=7.5$  mm plane. Once both waveguide ports are defined, the waveguide ports appear on the structure, as shown in the following figure:



## Setting up FEM Simulations

To set up an FEM simulation for a planar port:

1. Select **FEM Simulation** from the drop-down list available in the **Simulation** toolbar.
2. Click **Setup** (  ) to create and edit the setup of a new simulation. The **Setup FEM Simulation** window is displayed.
3. Add frequency plans.
4. Specify options in the **Mesh/Convergence Properties** tab.
5. In **Solver** tab, choose Direct solver.
6. Click **Done** in Setup FEM Simulation window.
7. Click  to perform the FEM simulation.

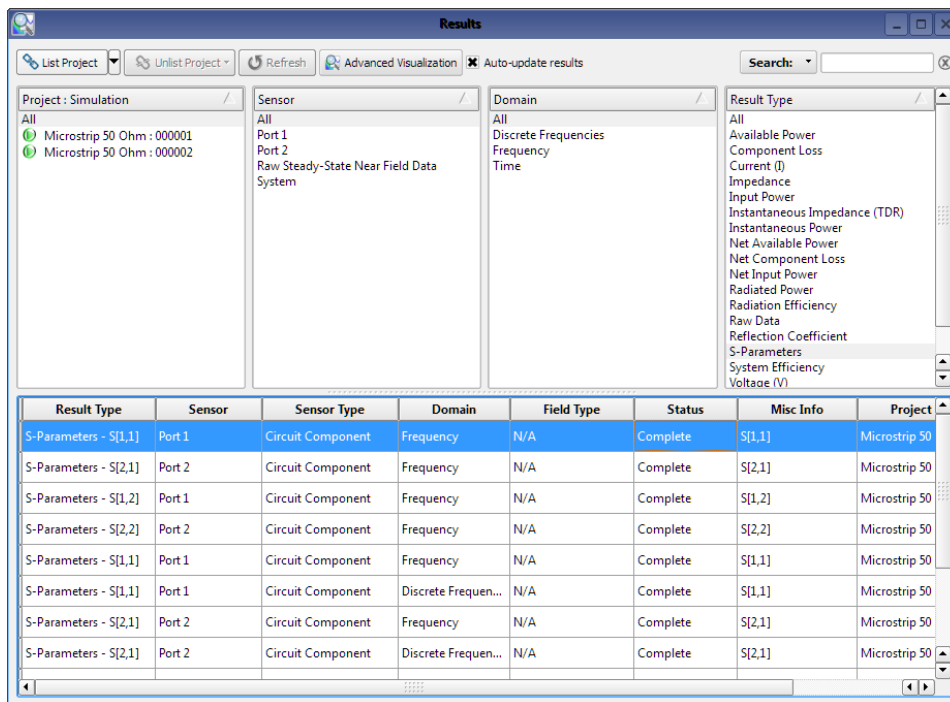
## Viewing Results

After simulating the microstrip line, you can view results in the Results window.

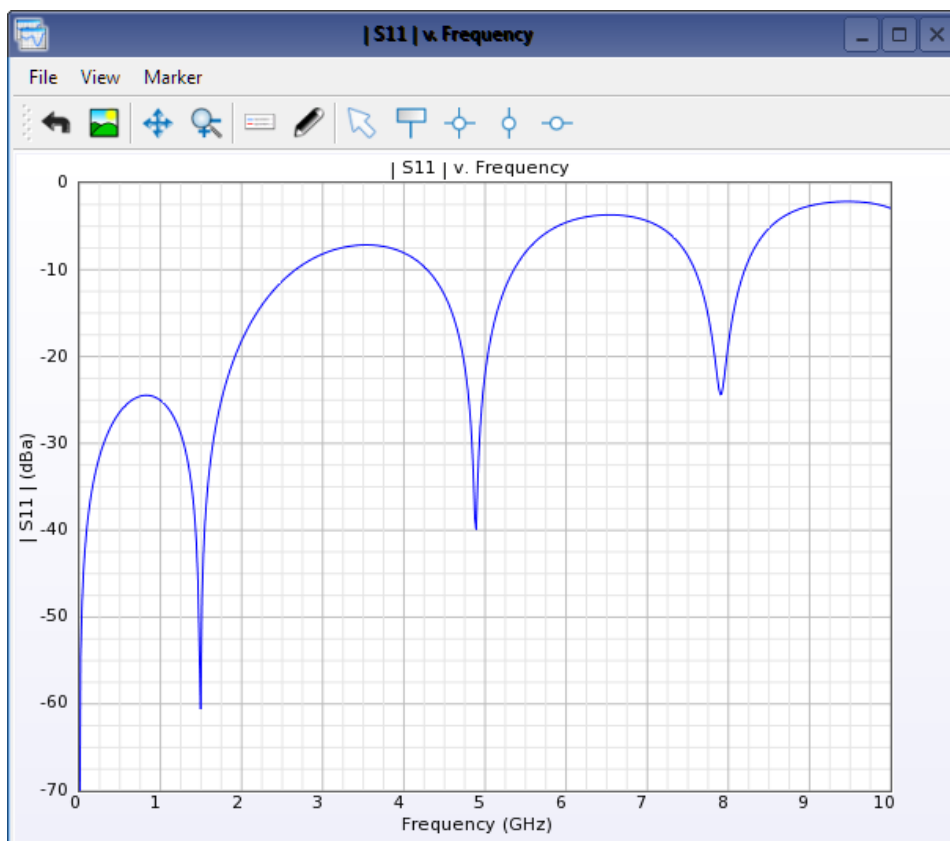
## Viewing Planar Results

To view results of a planar port simulation

1. Click **Results** to open the Results window. The results of FEM simulation are displayed.
2. Select **S-Parameters** in the **Result Type** column. The s-parameter results for port 1 and port 2 are displayed.



3. Double-click the required result type in the bottom pane to view a plot:



## Advanced Visualization

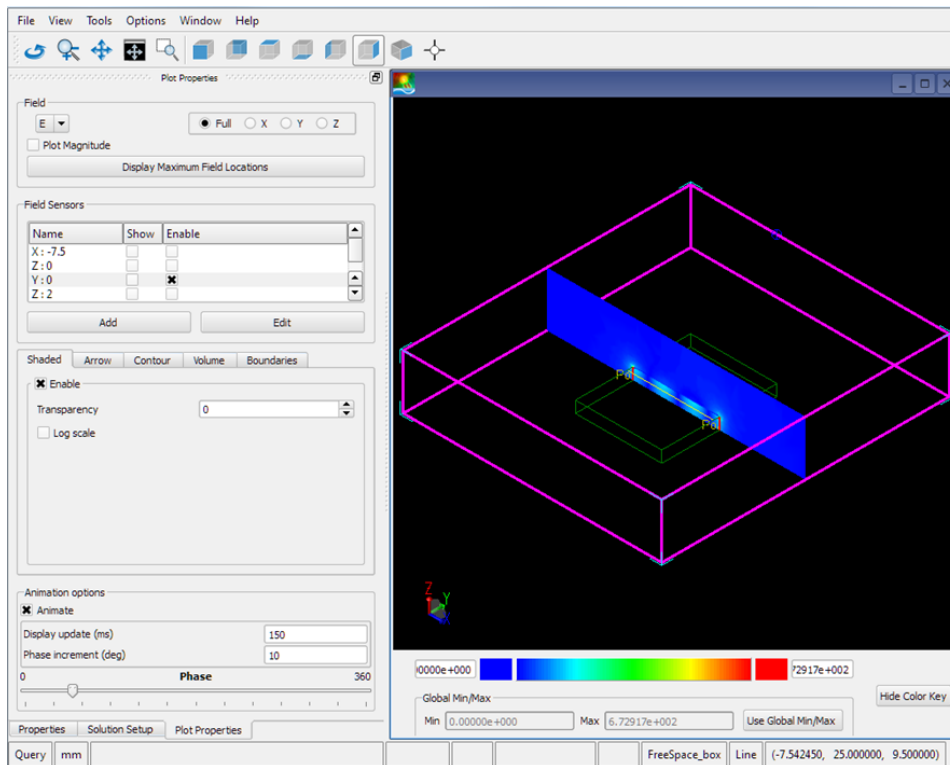
In the Advanced Visualization window, you can view boundaries, animation, E fields, and field sensors.

### NOTE

Animation is disabled for DC plots.

To view an animation:

1. Click **Advanced Visualization** in the Results window.
2. Click the **Plot Properties** tab.
3. In **Field Sensors**, select **Enable** for the required field sensor.
4. Select **Animate**. The animation options are enabled and progress of the field with phase is displayed in the Visualization window.
5. Specify a value in the **Display update (ms)** field.
6. Specify a value in the **Phase increment (deg)** field.



## Simulating a Microstrip Line with Sheet Port

In this example, a sheet current source is defined by performing the following tasks:

- Opening the Microstrip 50 Ohm Project.
- Adding Sheet to Port.
- Setting up an FEM Simulation.
- Comparing Results.

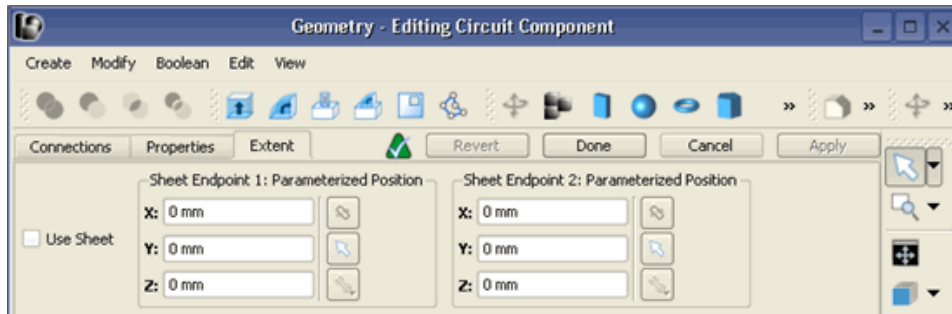
### Opening the Microstrip 50 Ohm Project


To open the Microstrip 50 Ohm line project:

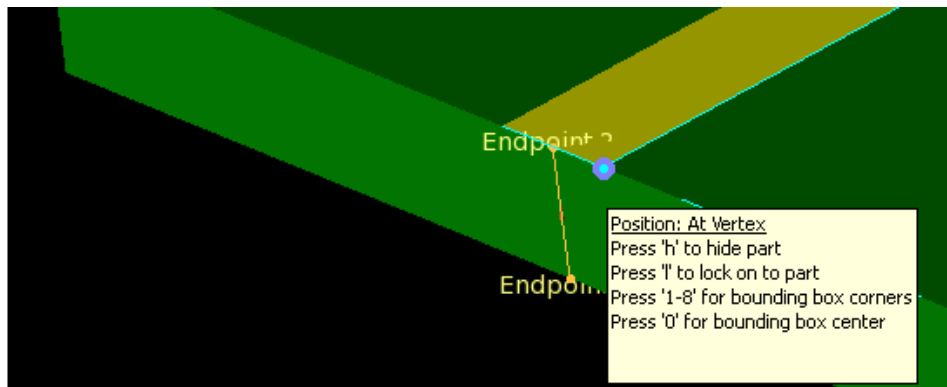
1. Select **Help > Example > Microstrip 50 Ohm**.
2. Select **File > Save** to save the project as **Microstrip\_50\_Ohm\_Sheet.ep**.

## Adding Sheet to Port

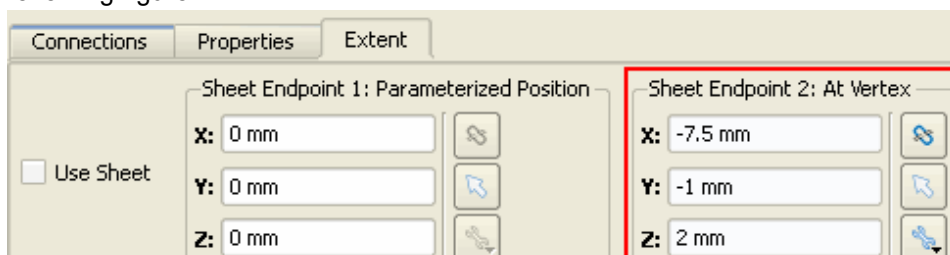
1. Double-click **Port 1** in the **Circuit Components/Ports** list. The Geometry-Editing Circuit Component window is displayed, as shown in the following figure:
2. Click the **Extent** tab. The sheet parameters are displayed.



3. For Endpoint 2, click  to specify the position and select one corner of Microstrip line as shown below:



Endpoint 2 displays the coordinates of one corner of the Microstrip line, as shown highlighted in the following figure:

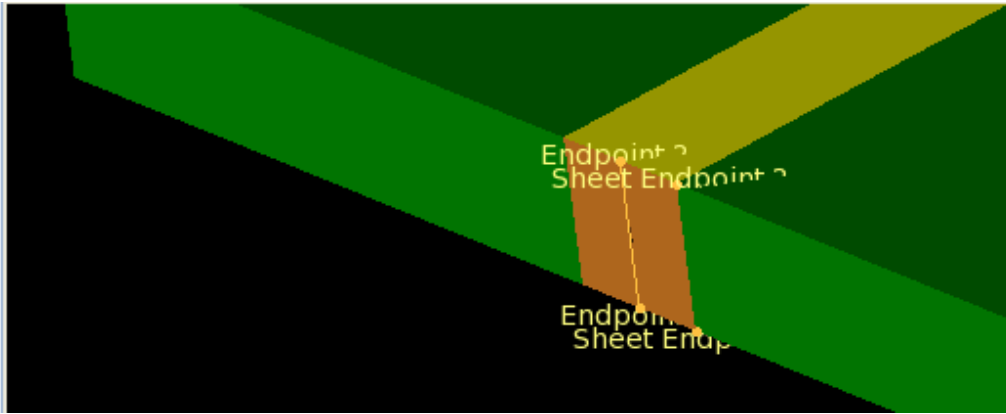


### NOTE

A sheet port needs to align with the polarity of a port. Otherwise, the sheet port becomes twisted.


4. Copy X and Y coordinates from Endpoint 2 and paste to Endpoint 1.
5. Select the **Use Sheet** check box.
6. Click **Done**.

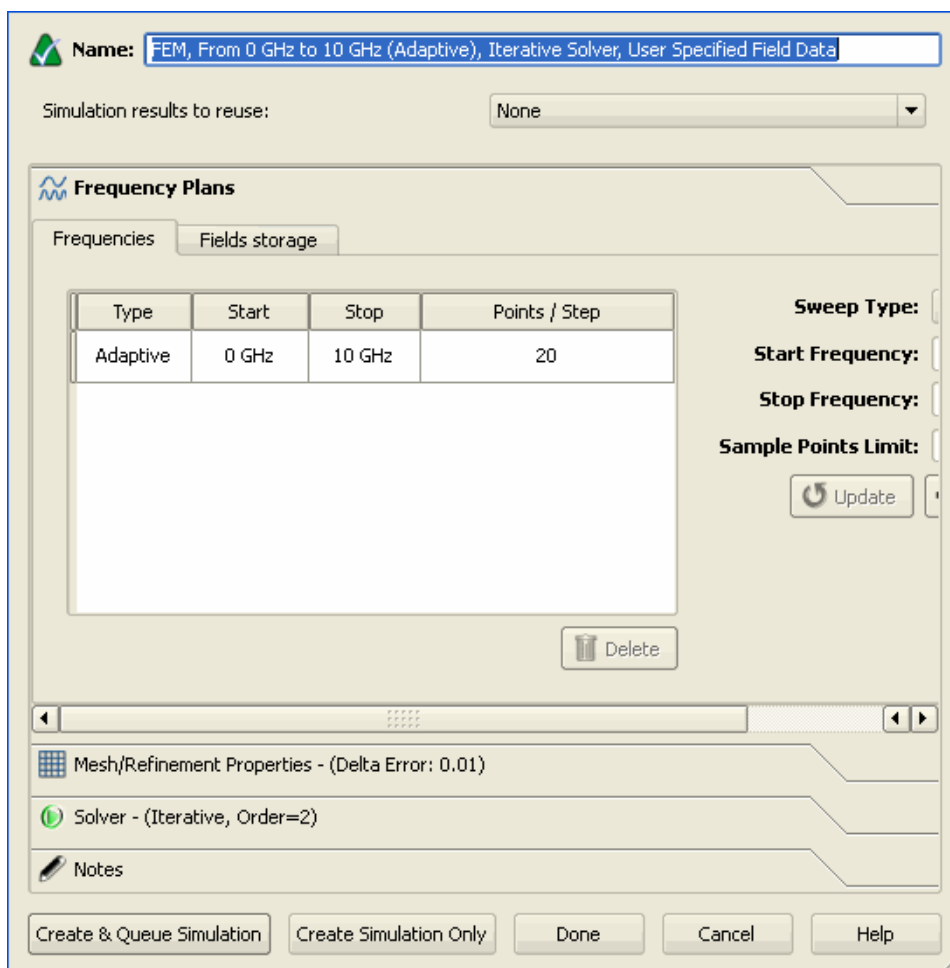
A sheet is created, which has width equal to the width of the Microstrip line starting from the base of the substrate to the Microstrip line. The following figure displays the required sheet:



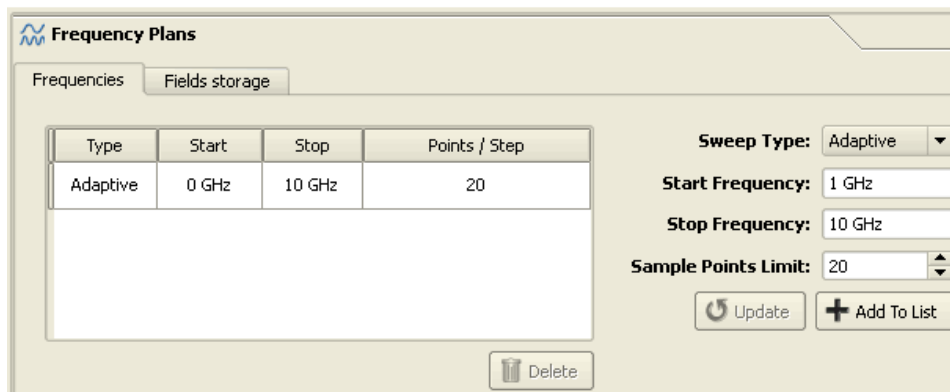
Similarly, add another sheet to Port 2.

## Setting up an FEM Simulation

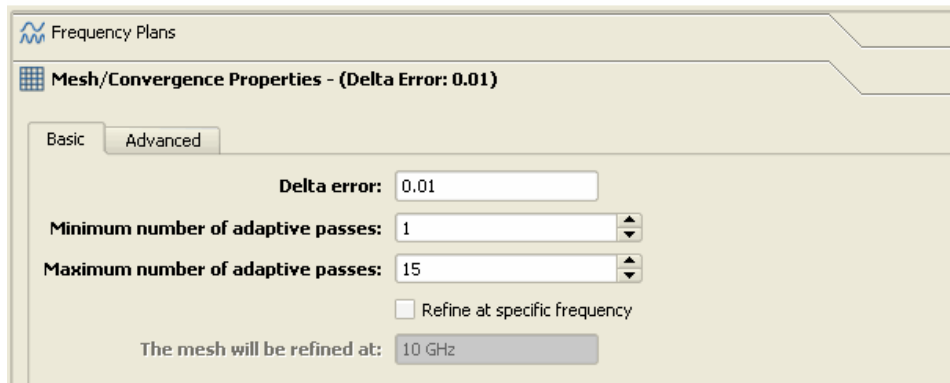
1. Select **FEM** from the drop-down list available in the **Simulation** toolbar.
2. Click **Setup** (  ) to create and edit the setup of a new simulation. The **Setup FEM Simulation** window is displayed, as shown in the following figure:



3. Add the frequency plans as shown in the following figure:



4. Click the **Field Storage** tab and select **User Specified frequencies**. This saves field data only at the starting and end frequency of the Adaptive frequency sweep.
5. Click **Setup Mesh/Refinement Properties**.
6. Specify the basic settings, as shown in the following figure:

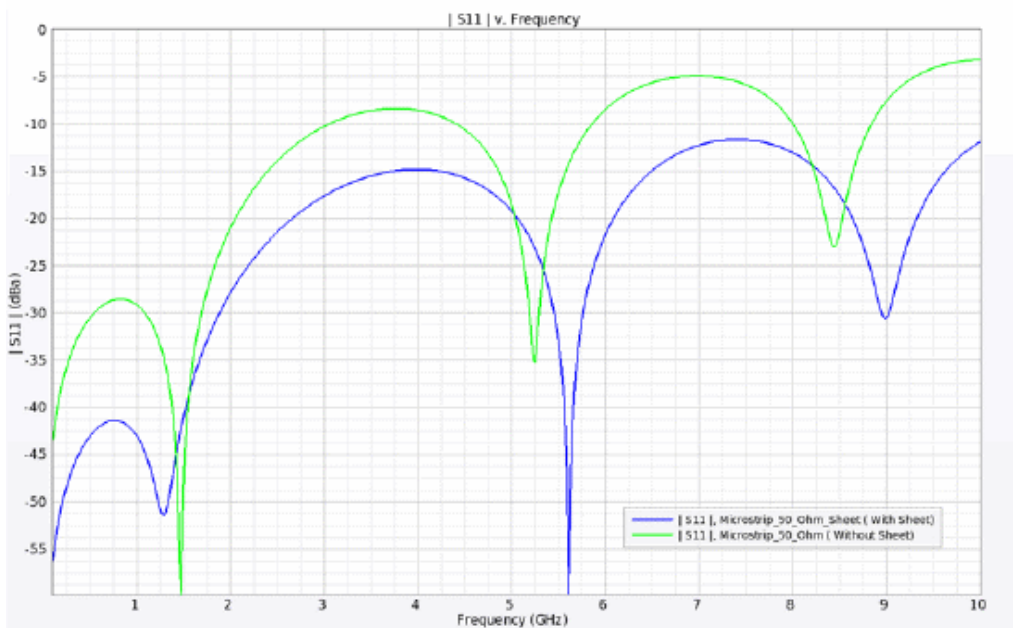


7. Click the **Advanced** tab and select **Use Initial Target Mesh Size**.
8. Click the **Matrix Solver** tab and select **Iterative**.
9. Click **Create and Queue Simulation** to start FEM simulation.

## Comparing Results

An important advantage of a sheet current source is that it has a smaller parasitic inductance than line current sources. This is highlighted in the following plot, showing S11 with and without sheet ports. If sheet ports are used, the resonance frequencies are shifted to the left. This shift is caused by the reduction of the parasitic inductance of the current sources.





## Simulating a Microstrip Line with Symmetric Plane

In this example, a Microstrip Line geometry with a symmetric plane is simulated by performing the following tasks:

- Opening the Microstrip 50 Ohm Line Project.
- Applying Symmetry Boundary Conditions.
- Setting up an FEM Simulation.
- Visualizing Symmetric Plane.

### Opening the Microstrip 50 Ohm Line Project

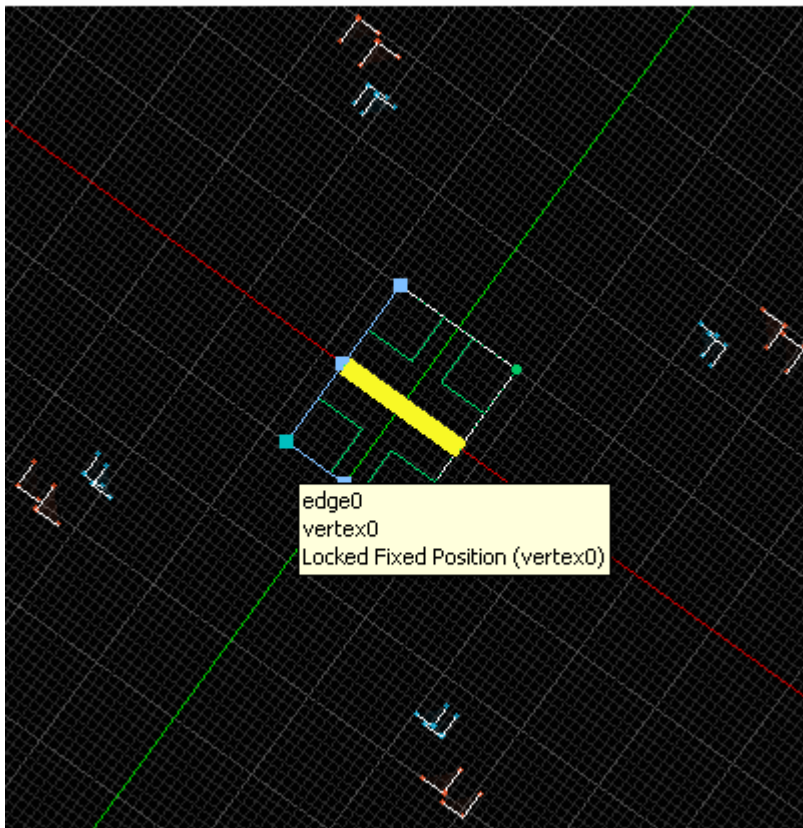
To demonstrate how to set up symmetric plane:

1. Select **Help > Example > Microstrip 50 Ohm line**.
2. Save the example as **Microstrip\_50\_Ohm\_Symm\_M.ep**.

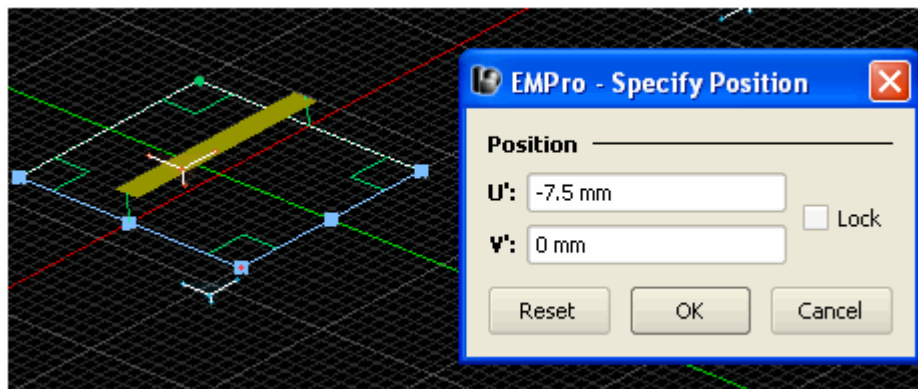
### Modifying Microstrip Line Geometry

To modify the Microstrip Line geometry:

1. Expand **Substrate** in the **Parts** list.
2. Click **Extrude** in the **Create Geometry** toolbar.
3. Select **Tools > Select/Manipulate** in the Geometry-Editing Extrude window.
4. Choose **Vertex 0** of the substrate for editing, as shown in the following figure:



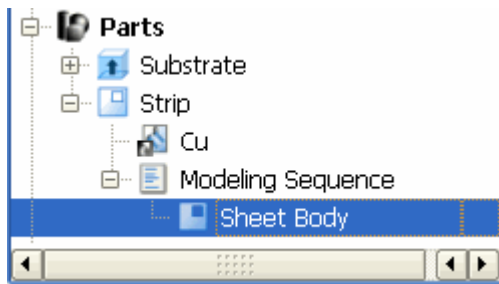
5. Right-click **Vertex 0** and select **Edit Position**, as shown in the following figure:



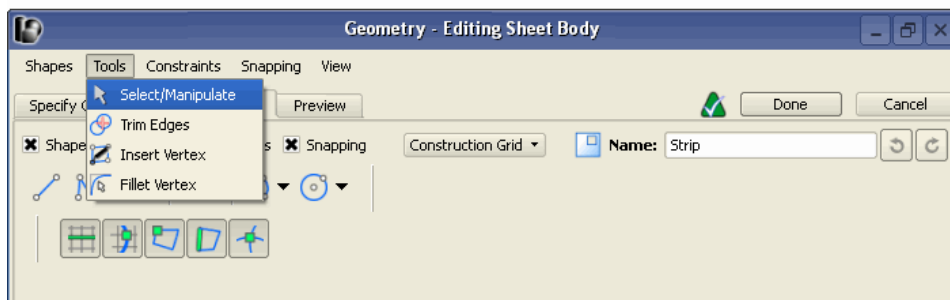
6. Change the value of **V** coordinate to **0**.
7. Clear the **Lock** check box to reduce the width of the substrate to half.
8. Click **OK**.
9. Click **Done** in the Geometry Editing Extrude window.

Similarly, to reduce the Strip width to half:

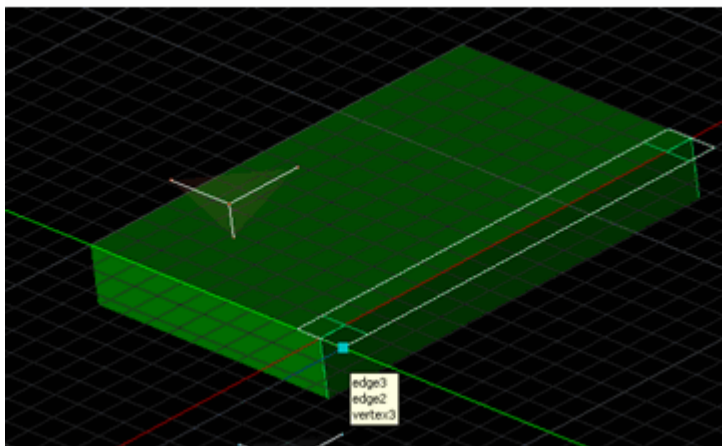
1. Expand **Strip** in the **Parts** list.



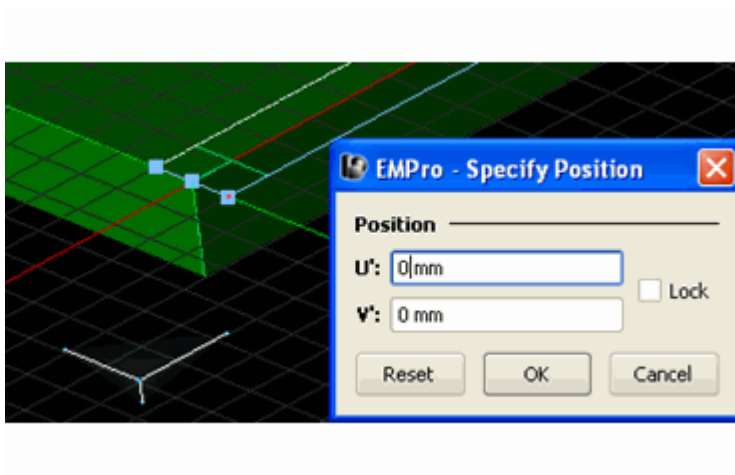
2. Double-click **Strip** to display the part geometry in the Geometry-Editing Sheet Body window.
3. Choose **Tools > Select/Manipulate** in the Geometry-Editing Sheet Body window, as shown in the following window:



4. Choose **Vertex 2** of the Strip for editing.

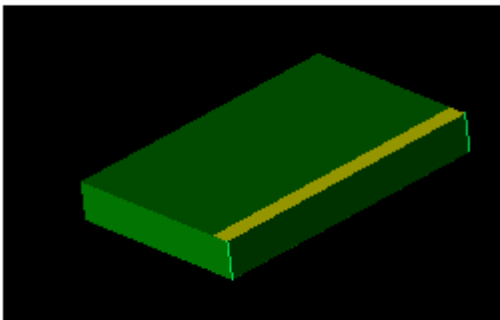


5. Right-click **Vertex 2** and select **Edit Position**. The Specify Position dialog box is displayed, as shown in the following figure:



6. Change the value of **V** coordinate to **0** and clear the **Lock** check box. This reduces the width of the Strip to half.
7. Click **Ok**.
8. Click **Done**.

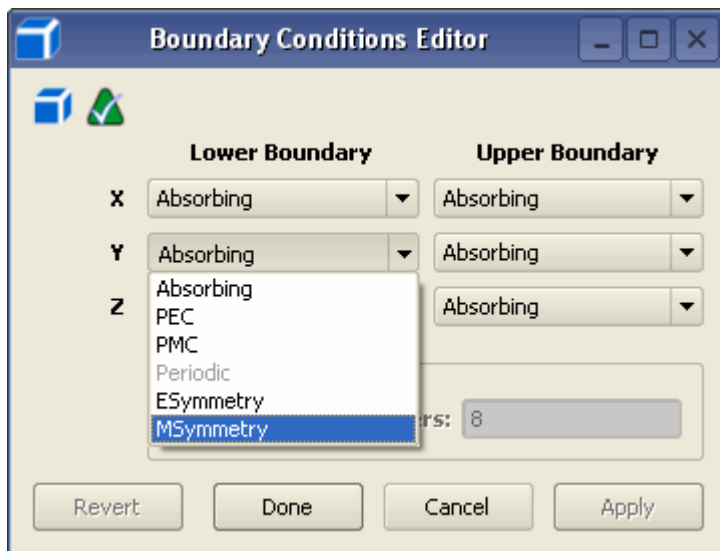
The ports are available at edges of the Substrate in the complete geometry, as shown in the following figure:



## Applying Symmetry Boundary Conditions

The Symmetry boundary condition is applicable only to the FEM simulator. To apply the symmetry boundary condition:

1. Choose **FEM** from the drop-down list in the Simulation toolbar.
2. Choose **Boundary Conditions** from **Simulation Domain** in the project tree. The Boundary Condition Editor dialog box is displayed.
3. In **Lower Boundary**, change the value of **Y** parameter to **MSymmetry**.




4. Click **Done**.
5. Select **FEM Padding** from the **Simulation Domain** list. The FEM Padding Editor dialog box is displayed.
6. Change **Y** parameter of **Lower** from **0 mm** from **20 mm**, as shown in the following figure:

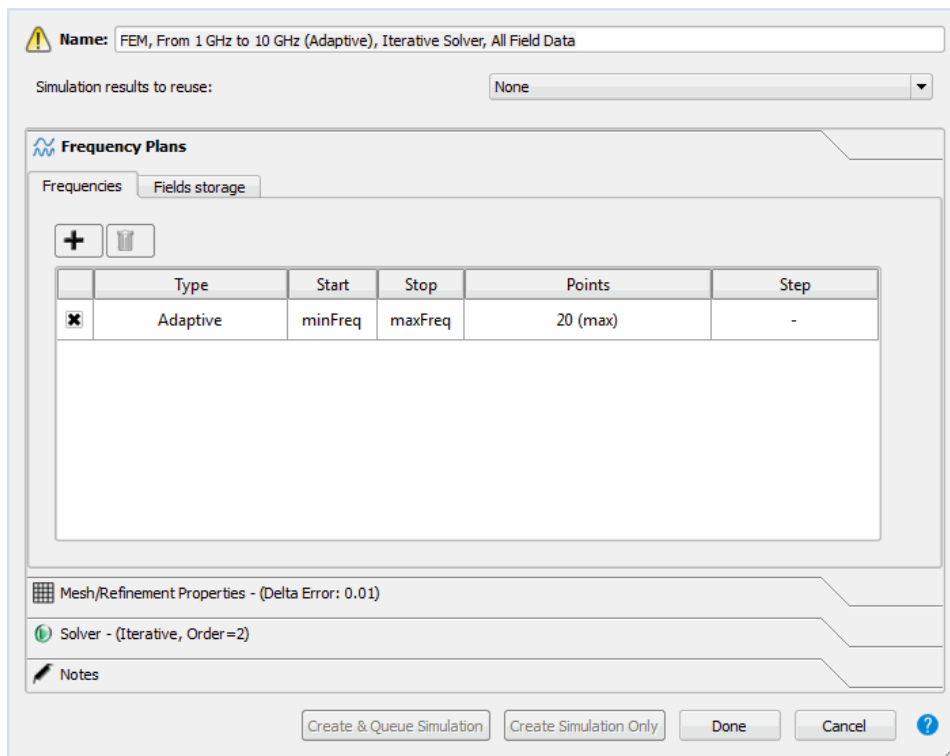


7. Click **Done**.

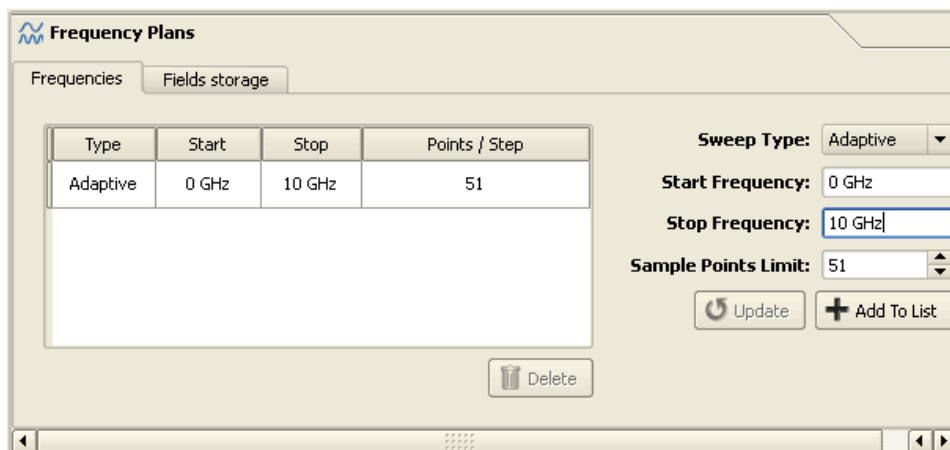
## Setting up an FEM Simulation

To set up an FEM simulation:

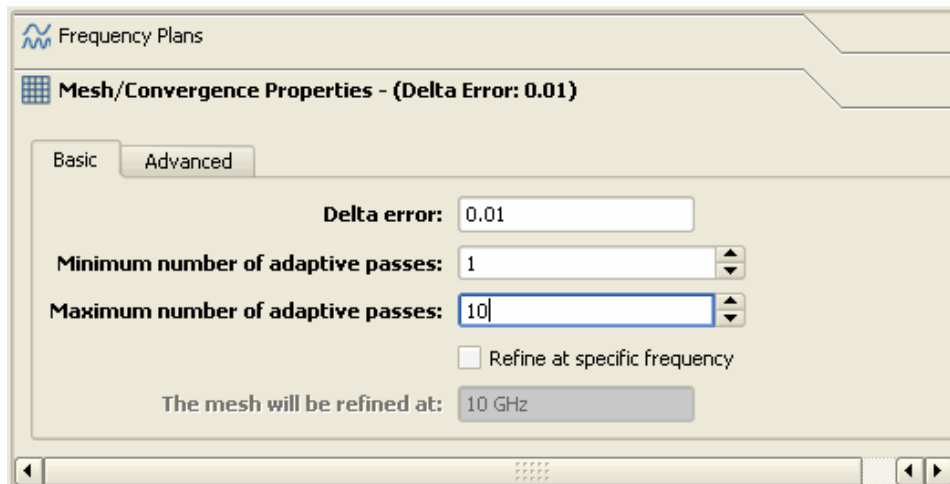
1. Select **FEM** from the drop-down list available in the **Simulation** toolbar.
2. Click **Setup** (  ) to create and edit the setup of a new simulation. The **Setup FEM Simulation** window is displayed, as shown in the following figure:



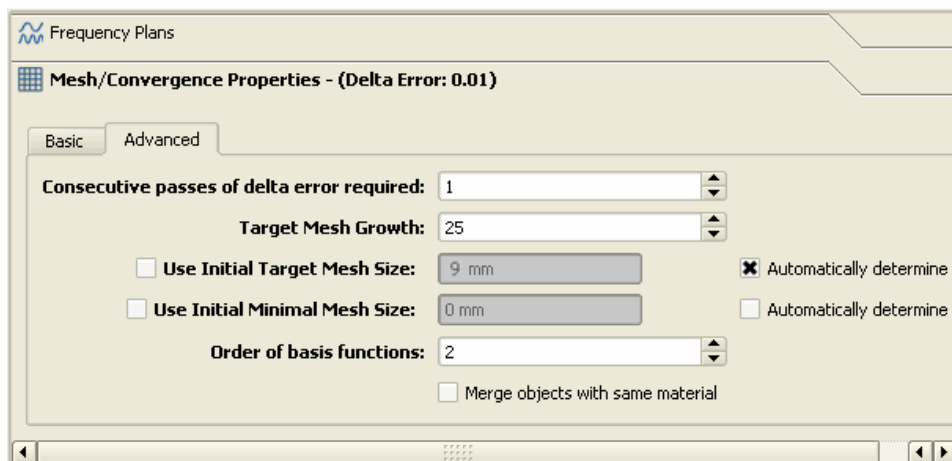
3. Add frequency plans, as shown in the following figure:



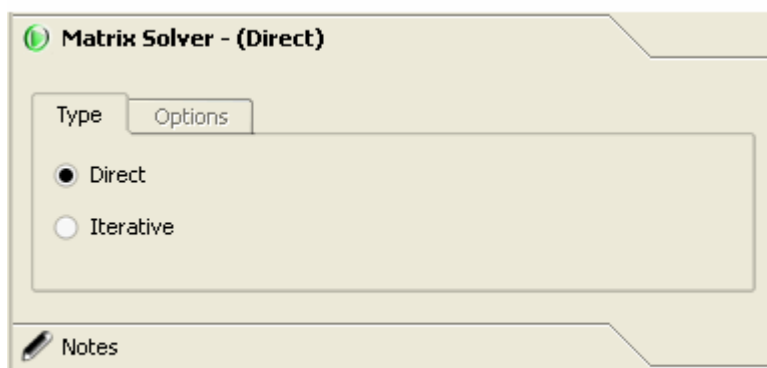
4. Click **Mesh/Refinement Properties** and specify the settings, as shown in the following figure:



5. Click the **Advanced** tab and specify the required settings, as shown in the following figure:




6. Click **Solver-(Direct, Order=2)**.  
7. Choose the **Direct** solver.

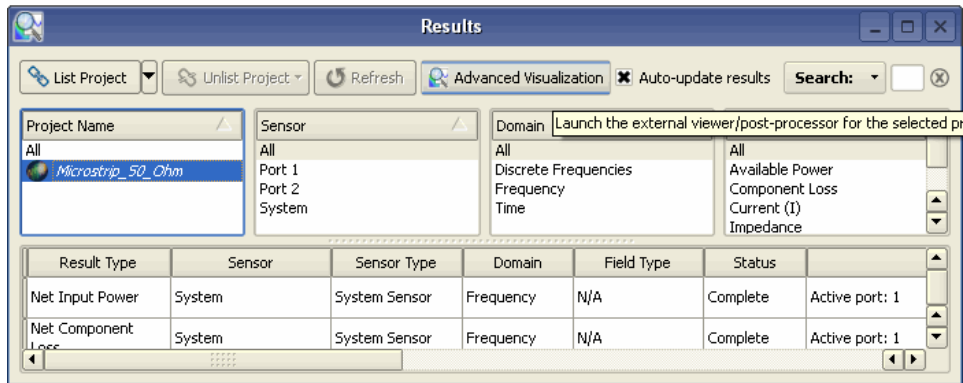


8. Click **Create and Queue Simulation** to start the FEM simulation.

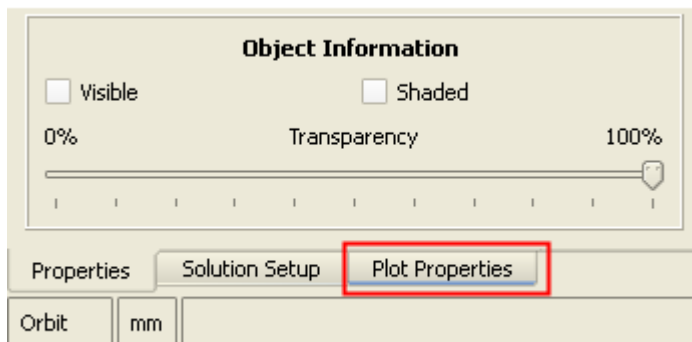
## Visualizing Symmetric Plane

To visualize the symmetric plane:

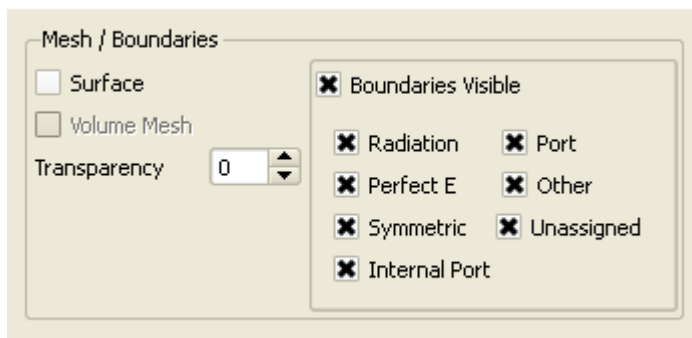
1. Click the **Results** tab  to open the Results window:



2. Select a project and click **Advance Visualization**.
3. In the **Advance Visualization** window, click the **Plot Properties** tab.

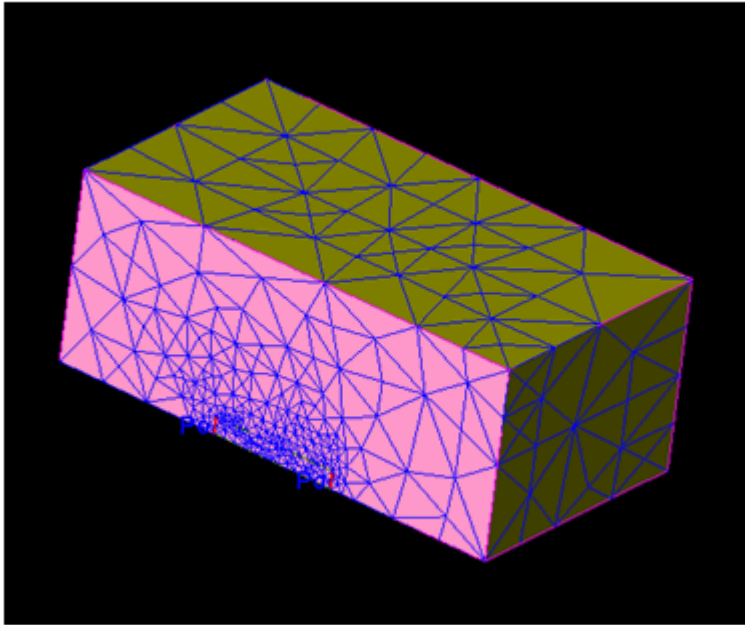


4. Select the **Boundaries Visible** check box in the **Mesh/Boundaries** pane.

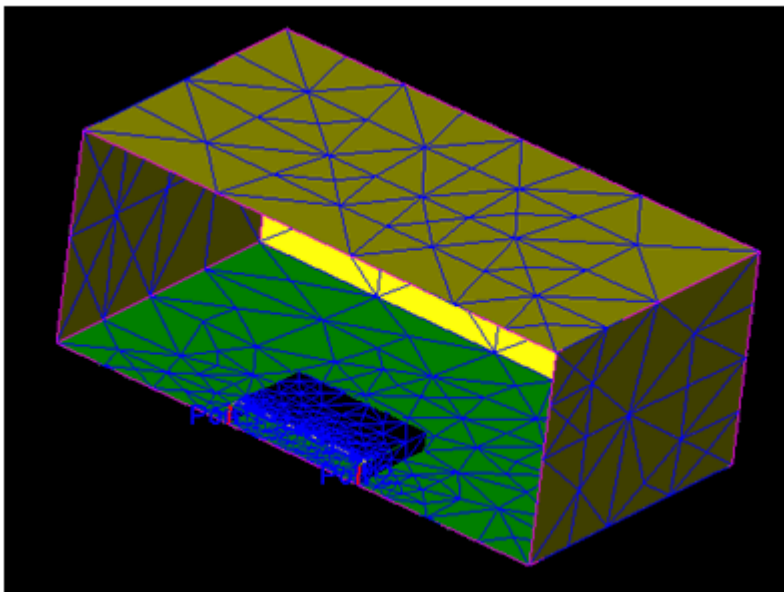


Boundaries are visible on the object, where the pink color represents the symmetric plane.





5. To view the internal structure of the object, clear the **Symmetric** check box in the Mesh/Boundaries pane.



## Performing Multimode Analysis on Rectangular Waveguide

In this example, a multimode analysis is performed on any structure in EMPro. To demonstrate the design flow, a square waveguide is used in this example.

## Project Setup

- Create a waveguide geometry
- Simulate a waveguide model

## Simulation Setup

- Simulation Engine: FEM
- Simulation frequencies and sweep: 0 to 10 GHz and Adaptive Freq Sweep
- Simulation Accuracy (Delta-S): 0.02 (2%)
- Solver: "Direct Solver"

## Results



- View S Parameter
- View Propagation Constant
- View Field

After you have created a design, you need to complete the following procedure before performing a simulation. However, you do not need to perform these steps sequentially.

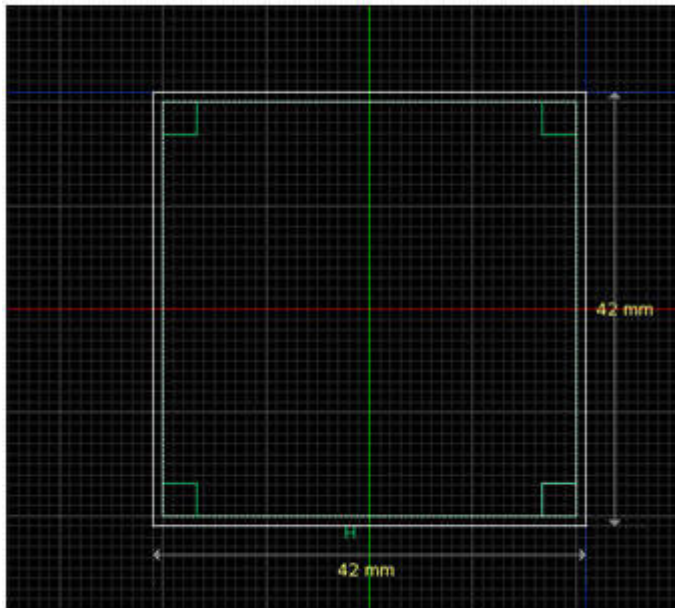
## Step 1: Setting up Waveguide Geometry Model

You can create a waveguide geometry in the Geometry window by using the geometry tools.

To create a waveguide:

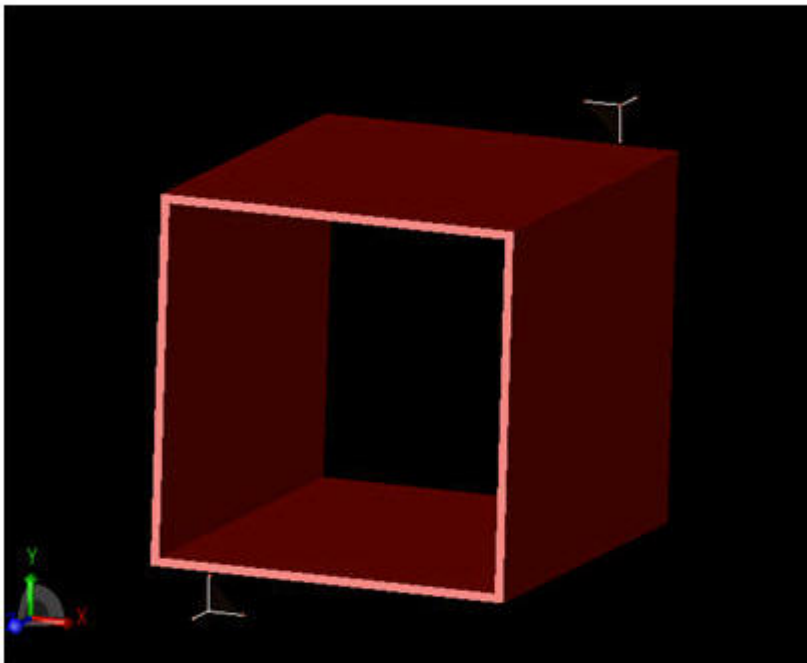
1. Click **Extrude** (  ) on the **Create Geometry** toolbar.
2. Select the **Rectangle** tool  and specify the inner dimensions as ( -20 mm, -20 mm) and (20 mm,20 mm). Also, specify the outer dimensions as (-21mm, -21 mm) and (21 mm,21mm).

3. Extrude the waveguide to 50 mm length. This will draw a square waveguide of 40 mm x 40 mm cross section and length 50 mm.

**NOTE**

This waveguide supports TE<sub>10</sub> and TE<sub>01</sub> mode at C band.

4. Specify the name the of the waveguide as **Square Waveguide** and assign the material **Aluminium**. The resulting waveguide appears as follows:

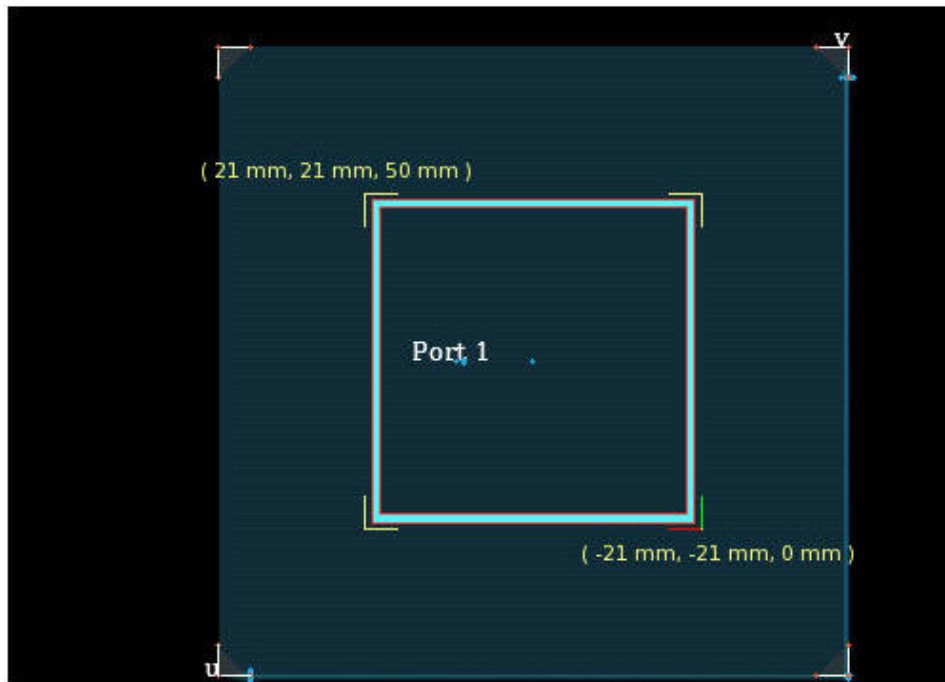


## Setting up Waveguide Ports

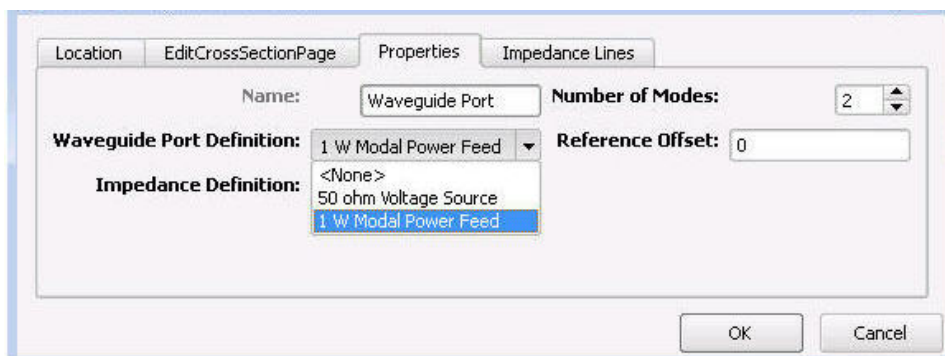
You can set up a waveguide port by performing the following steps:

1. Select **FEM Simulation** from the drop-down list available in the **Simulation** toolbar.
2. Right-click **Circuit Components/Ports** and select **New Waveguide Port** to a new waveguide port. This opens the **EMPro Waveguide Port Editor**.
3. Choose view **Bottom**( +z)to define the Waveguide port at Z equals to 0mm value. From **Waveguide Port Editor** >

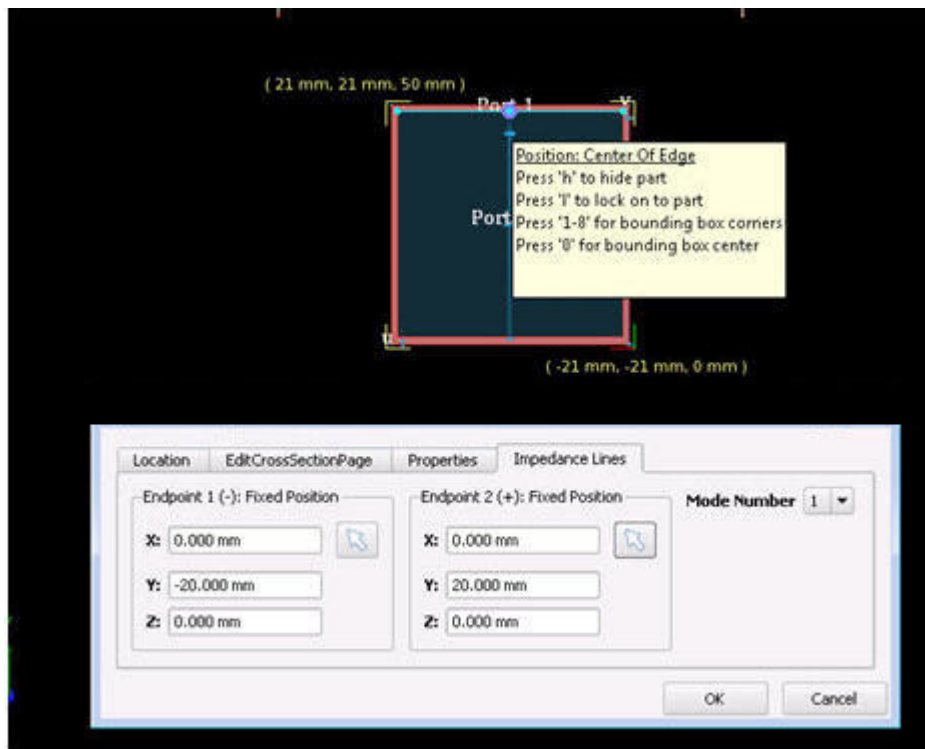
**Location**, choose  to select the face:



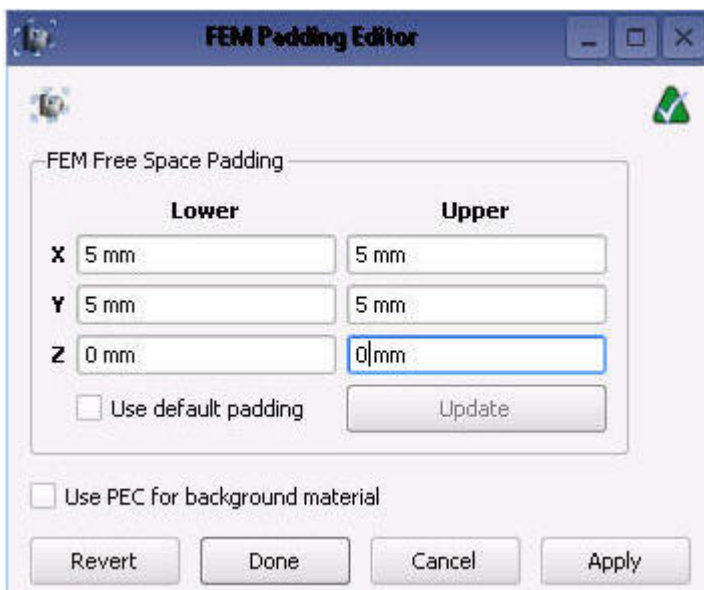
4. In the **EditCrossSectionPage** tab, remove the selection of **Auto Extend to simulation domain boundaries** to restrict the waveguide port to waveguide cross section only.
5. In the **Properties** tab, choose **Waveguide Port Definition** and choose the number of modes equals to 2. Also, for Waveguide port definition choose 1 W Modal power feed, as shown in the following figure:



6. In the **Impedance Lines** tab, choose **center of the lower edge** from **Impedance for Endpoint 1**.
7. For endpoint 2, choose **center point of the upper edge**.



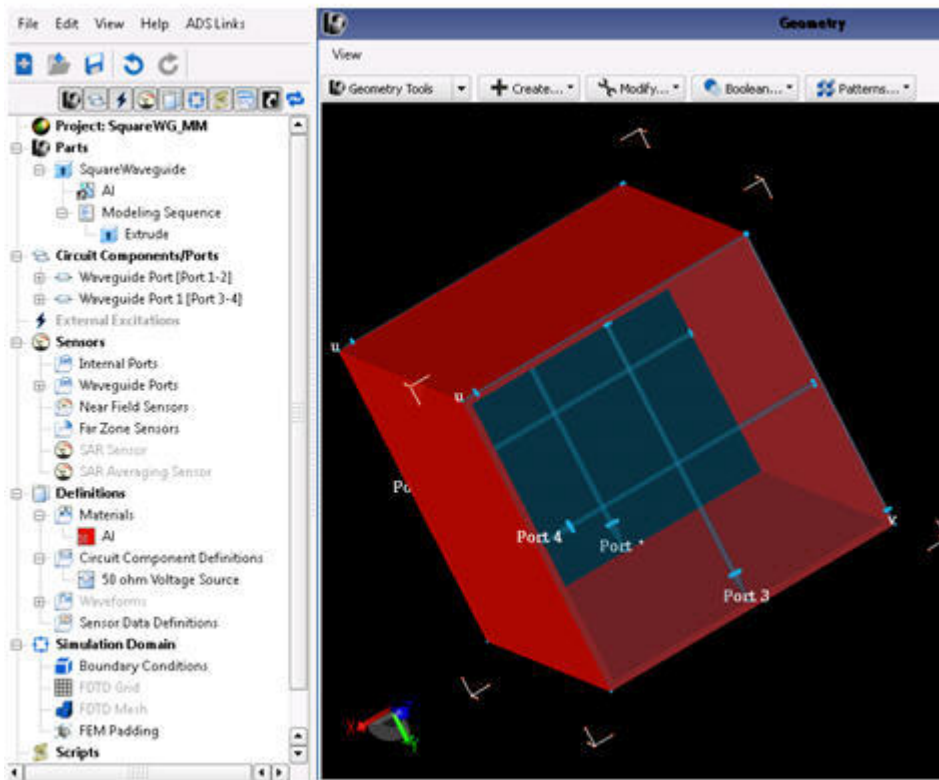
This defines mode 1 for port 1. Similarly, you can define mode 2 for port 1. If Waveguide port 1 is displaying invalid sign, change the padding as shown in the following figure:




While specifying the FEM padding of waveguide structure, consider the following guidelines:

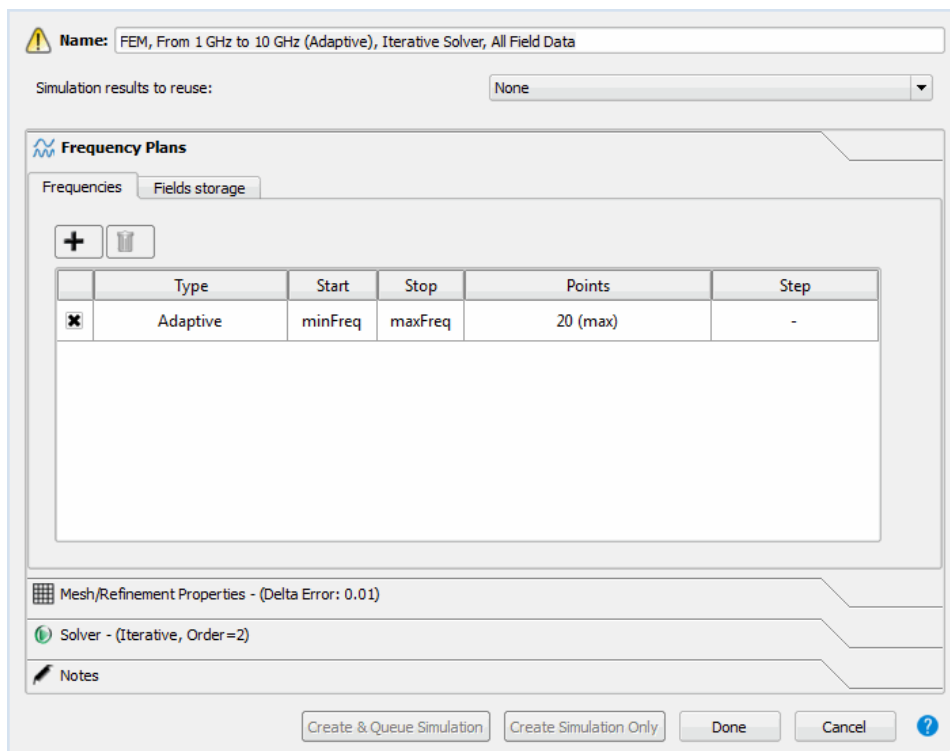
- The +/- Z padding MUST be zero for a valid waveguide port definition. The padding in the orthogonal direction of waveguide plane should be 0 mm. In this case, the Z axis is orthogonal to waveguide planes. After placing these paddings, the waveguide port will become valid.
- The +/- X and Y padding COULD be set to zero for a more efficient simulation Choose Top(-z) view to define second waveguide port at z=50 mm. Repeat the same steps as defined above to create second waveguide port and two modes.

A Waveguide with waveguide ports having two excited modes appears as follows:



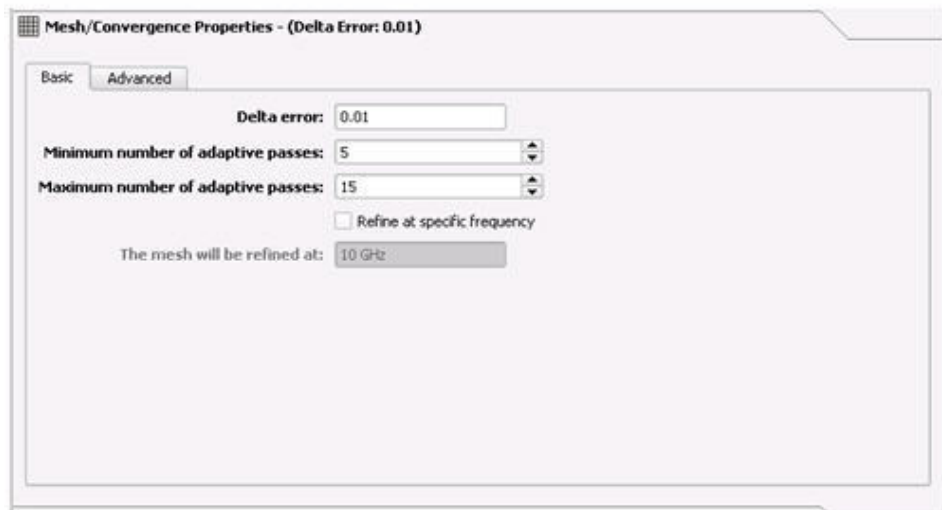
## Setting up Simulation

1. Select **FEM** from the drop-down list available in the **Simulation** toolbar.
2. Click **Setup** (  ) to create and edit the setup of a new simulation. The **Setup FEM Simulation** window is displayed, as shown in the following figure:



3. Define the frequency plan.

4. Click **Setup Mesh/Refinement Properties**.
5. Choose the mesh convergence properties, as shown in the following figure:

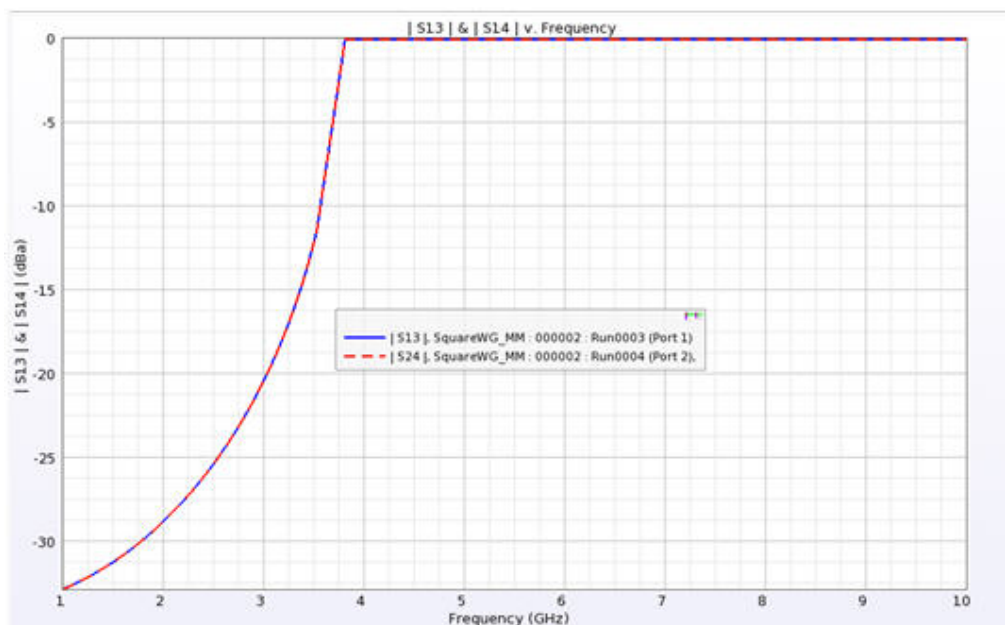


6. Click the **Matrix Solver** tab and select **Iterative**.
7. Click **Create and Queue simulation**.

## Viewing Results

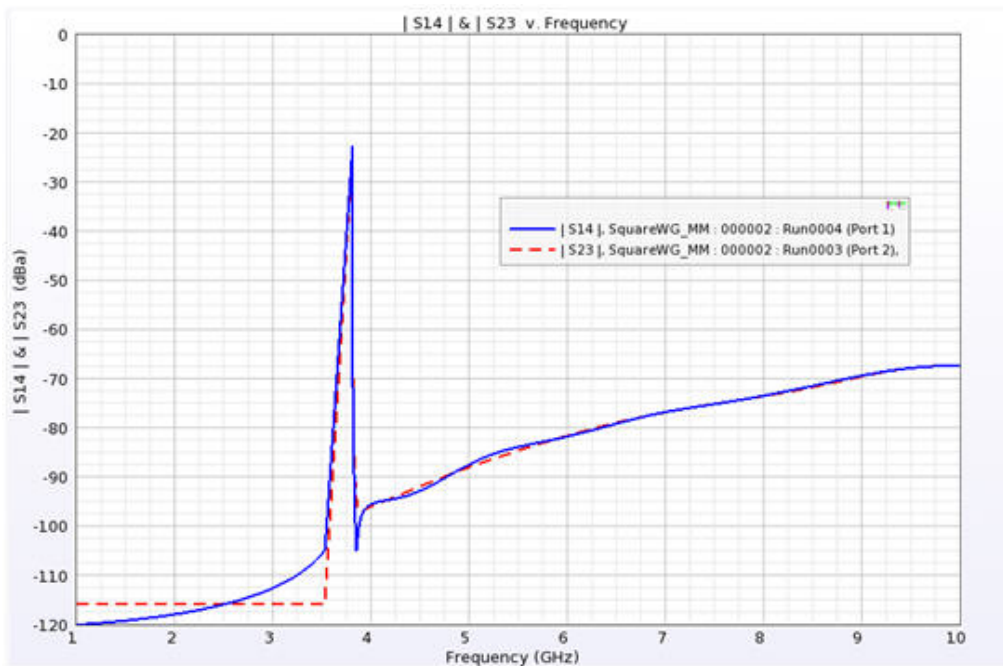
### Viewing S Parameter

For waveguide port 1, port 1 and port 2 are defined for two modes. For the waveguide port 2, port3 and port4 are defined for two modes. Therefore, port 1 of waveguide port 1 will couple to port 3 of waveguide port2 since they are for same mode. Similar coupling takes place for port 2 and port 4. The S13 and S24 plot is displayed in the following figure:



Cross mode coupling S14 and S23 is displayed in the following figure:

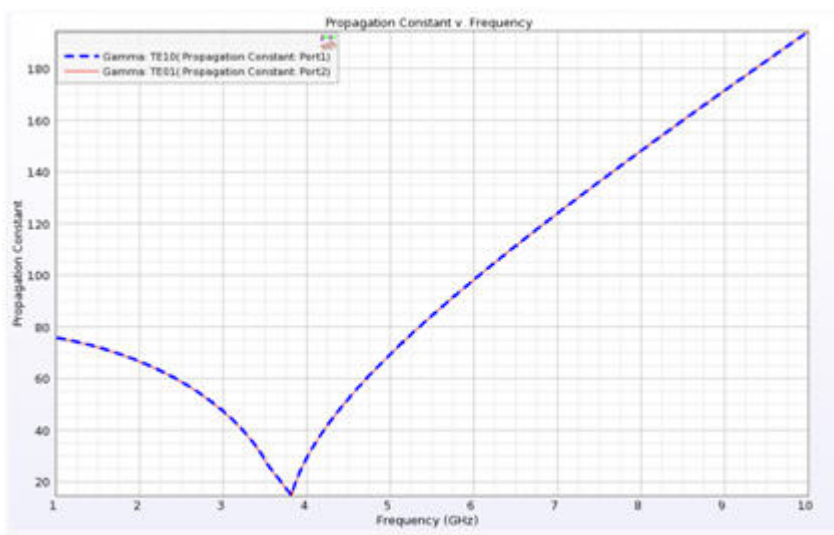




The set of equations solved by FEM are without any condition at the waveguide mode cut-off frequencies. This leads to the spike in S14 and S23 seen around 3.75 GHz

## Viewing Propagation Constant

TE<sub>10</sub> and TE<sub>01</sub> mode is degenerate mode in square waveguide. For the degenerate modes, the propagation constant is same as shown in plot, as displayed in the following figure::

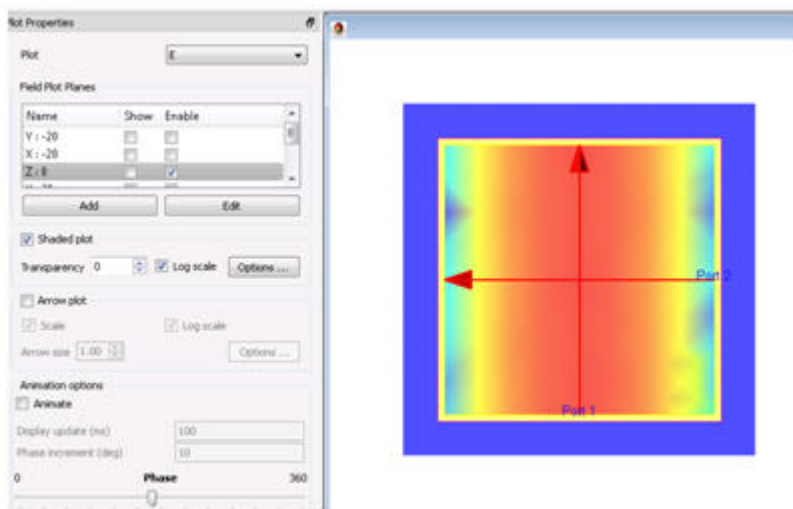


## Viewing Field

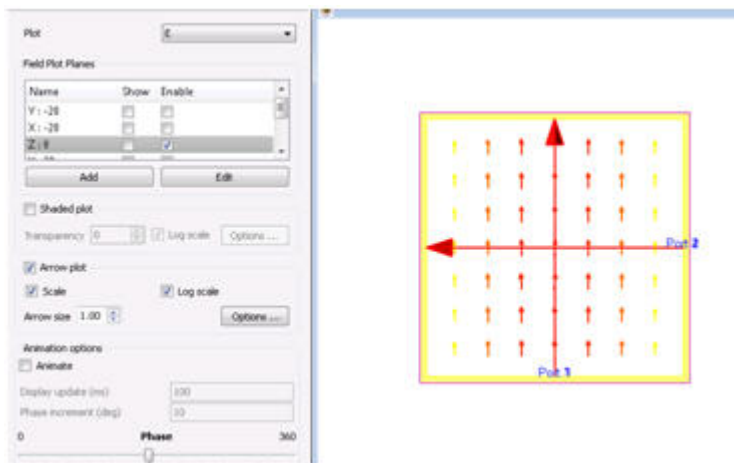
You can viewing fields for different modes by selecting a project from the Results window and clicking **Advance Visualization**. In the Advance Visualization window, click Solution Setup and select one of the frequency in pass band. For the following plots, 6.9 GHz frequency is selected. Also, choose Port 1 mode 1 to see TE<sub>10</sub> mode in the waveguide.

In the Plot Properties tab, choose Z=0 in the field plot planes. Choose Shaded plot and log scale. The plot will appear as shown in the following figure:

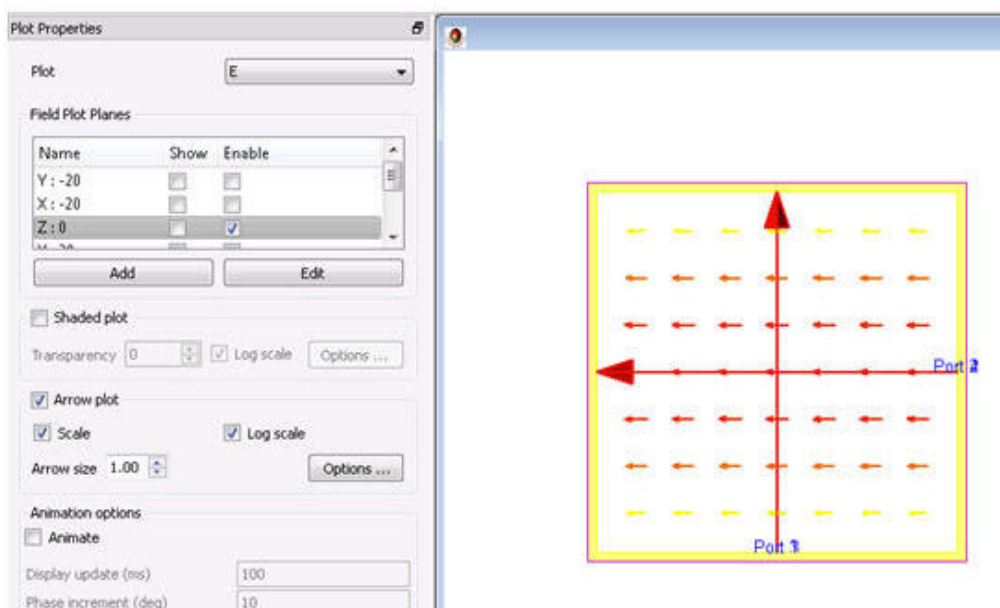




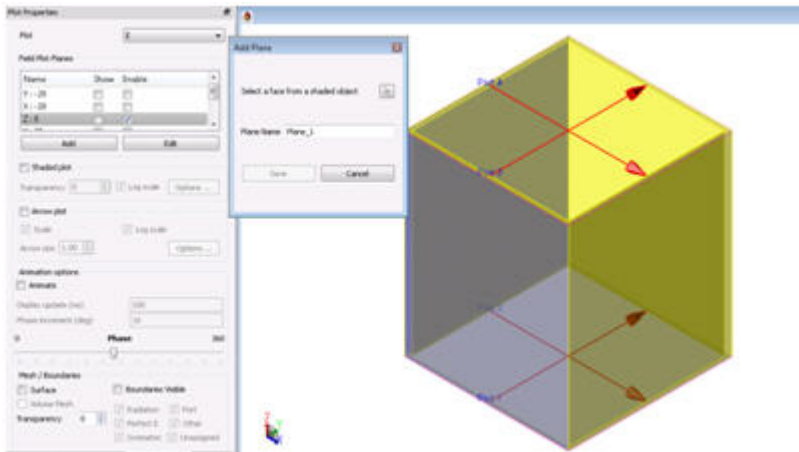
Select **Animate** to see the progress of the field with phase. Remove the selection from **Shaded plot** and select **Arrow plot**. This is TE<sub>01</sub> mode, as shown in the following figure:



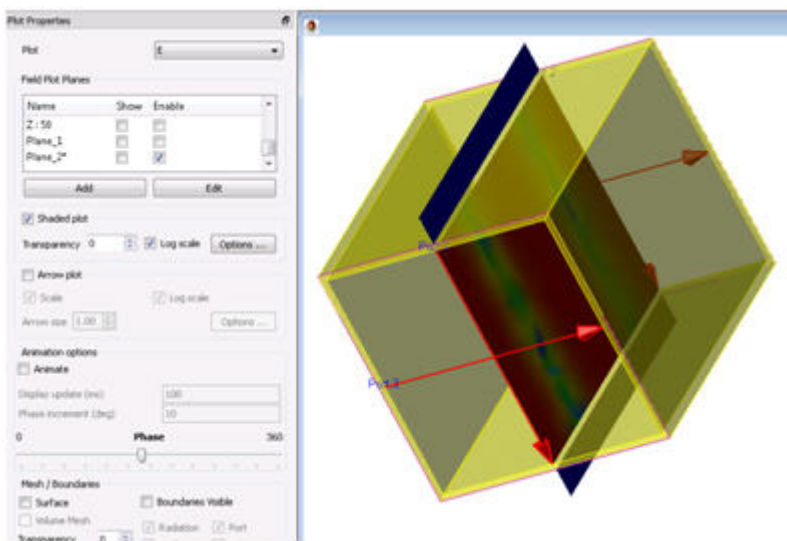
Click the **Solution Setup** tab and select **Port 1 mode2**. Choose shaded plot in solution setup. Clear the **Shaded Plot** check box and select the **Arrow Plot** check box. The following figure displays the TE<sub>01</sub> mode:



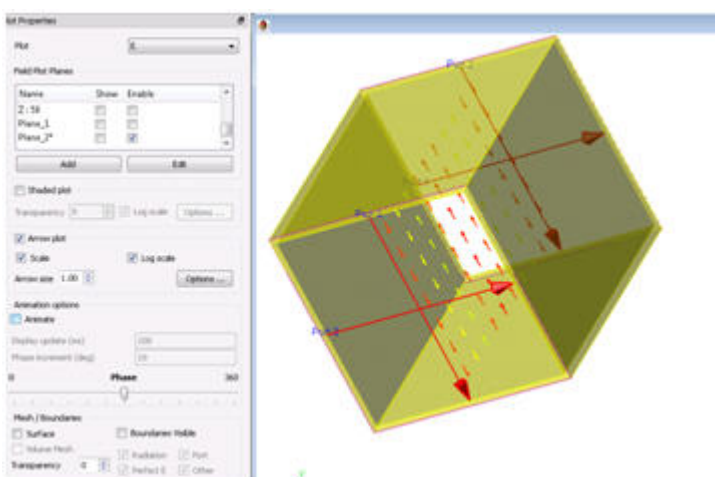
Add a plane by clicking **Add** in field plot planes to see the field along the length of the waveguide section.



Move this plane on the center of the waveguide by using edit option. Choose Shade plot to see fields on this plane.



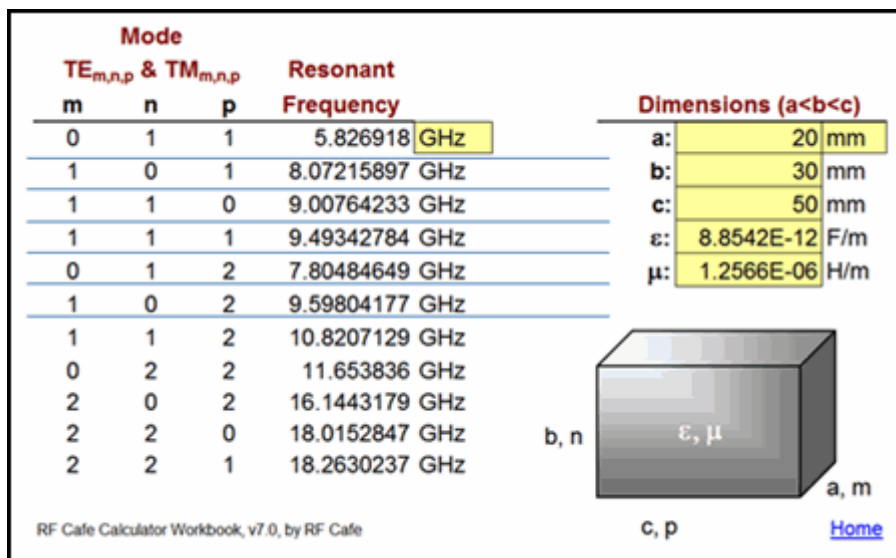
Similarly, arrow plot on the plane is shown below:



# Performing an Eigenmode Simulation for a Rectangular Cavity

In this example, an eigenmode simulation is created by using a rectangular cavity. The theoretical resonant frequencies are generated by RF Cafe Calculator assuming PEC walls and vacuum inside the rectangular cavity.

The following figure displays the dimensions and analytical resonant frequencies used in this example:



Perform the following tasks to create an Eigenmode simulation for the rectangular cavity:

- [Creating a New Project](#)
- [Creating a New Geometry](#)
- [Creating and Assigning Materials](#)
- [Defining the Outer Boundary](#)
- [Setting up an Eigenmode Simulation](#)
- [Running the Eigenmode Simulation](#)
- [Viewing Eigenmode Simulation Results](#)

## Creating a New Project

To create a rectangular cavity:

1. Open a new project.
2. Select **Edit > Project Properties Editor** to open the Project Properties Editor window.
3. Select the **Display Units** tab in the Project Properties Editor window.
4. Select **SI Metric** in the Unit Set drop-down list.
5. Change **Length** to millimeters (mm). This changes the Unit Set value to **Custom**.
6. Change **Frequency Unit** to **GHz**.

7. Click **Done**.

## Creating a New Geometry

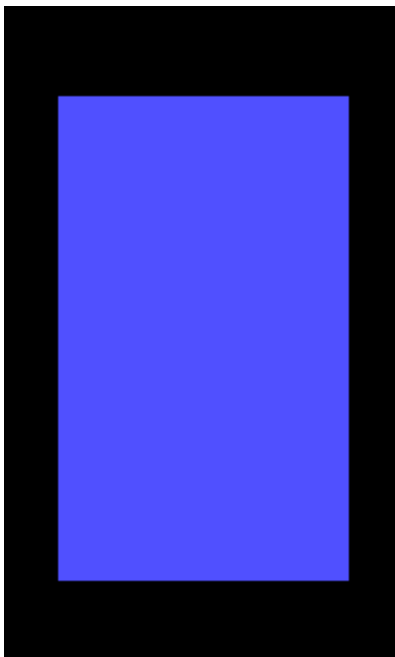
After creating a new project, you can add a new geometry by performing the following steps:

1. Right-click **Parts** and choose **Create New > Extrude**.
2. Choose the **Rectangle** tool from the **Shapes** toolbar.
3. Type a value in the **Width**, **Depth**, and **Height** text box.
4. Choose the **Specify Orientation** tab and modify the parameters according to your requirements.
5. Click **Done**. A rectangular box is visible in the Geometry window.

## Creating and Assigning Materials

To assign a material to the rectangular box:

1. Right-click **Materials** in the **Definitions** list and **Select from Default Material Library**.
2. Select **Air** in the **Add a Default Material** window.
3. Click **Add**. The Air material is displayed in the **Materials** list.
4. Click and drag the **Air** material object and drop it on top of the rectangular box object present in the **Parts** list. The following figure displays the rectangular box after assigning the Air material object.



### NOTE

You do not need to specify excitations in the Eigenmode solver. You can skip this step. Excitations in Eigenmode solver may abort a simulation.

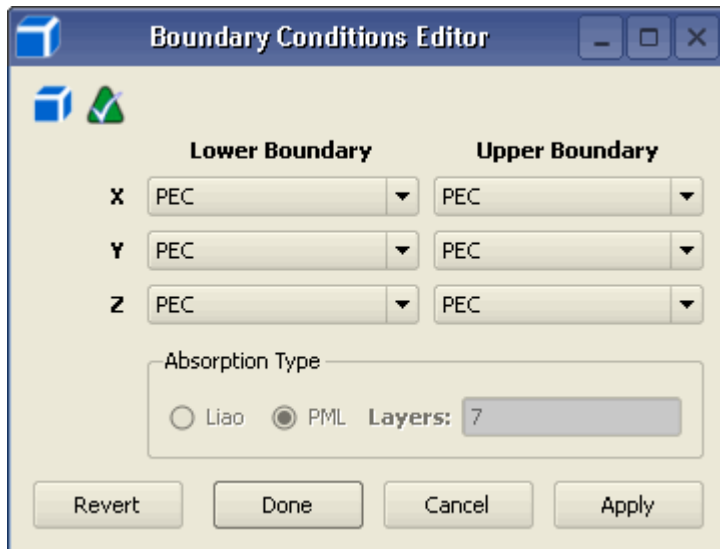
## Defining the Outer Boundary

The Eigenmode solver supports only closed structures. A closed structure is of the following types:

- A structure with perfect conducting boundaries: PEC or PMC. In this release, the Eigenmode solver does not support Radiation, Esymmetry, and Msymmetry conditions.
- A structure with metal surroundings, such as a copper box or an aluminum cylinder. For more information, refer [#Adding Lossy Metal](#).

To define an outer boundary:

1. Double-click **Boundary Conditions** in the **Simulation Domain** list. The **Boundary Conditions Editor** dialog box is displayed.
2. Set the outer boundary properties to **PEC** for all boundaries.



3. Click **Done** to apply the outer boundary settings.
4. Double-click **FEM Padding** in the **Simulation Domain** list. The **FEM Padding Editor** dialog box is displayed.
5. Set the **Upper** and **Lower** limits to 0, as shown in the following figure:



#### NOTE

It is important to specify zero value for FEM padding. If you keep the default FEM padding values 20 mm, the 20 x 30 x 50 cavity would be expanded to 60 x 70 x 90. It is recommended to use zero padding for all Eigenmode simulations.

## Setting up an Eigenmode Simulation

To define an **Eigenmode simulation setup**, you need to specify frequency plans and mesh refinement options.

## Running the Eigenmode Simulation

After completing Eigenmode Simulation setup, you can run calculations on the geometry. The Simulations workspace window stores the project simulation(s). You can create, queue, and run simulations using the Simulations workspace.

To run an Eigenmode simulation, click **Create and Queue** in the Simulations workspace.

## Viewing Eigenmode Simulation Results

To make the cavity lossy and observe changes on the Q values, you can add lossy dielectric and metal.

## Adding Lossy Dielectric

Replace the air by a lossy dielectric with loss tangent  $\delta = 0.001$  and dielectric constant 1. Theoretically, the Q value is equal to the following equation:

$$1/\delta = 1000$$

The output of Eigenmode solver, Q values are 1000 for material with tangent 0.001, is displayed in the following figure:

Eigenfrequencies		Q values
1.	5.827329e+009	1.000000e+003
2.	7.807202e+009	1.000000e+003
3.	8.074515e+009	1.000000e+003
4.	9.013233e+009	1.000000e+003
5.	9.497752e+009	1.000000e+003
6.	9.500686e+009	1.000000e+003
7.	9.602118e+009	1.000000e+003
8.	1.029852e+010	1.000000e+003
9.	1.043970e+010	1.000000e+003
10.	1.082674e+010	1.000000e+003

The dielectric material with conductivity can be put inside the cavity and the Q values close to theoretical results are produced, which is not shown in this document.

## Adding Lossy Metal

You need to build copper walls for a vacuum cavity with the same size. Select **Modify > Shell** and set **Specify Thickness** value to 1 mm. Copper is assigned to this shell and inside the shell it is vacuum. In this example, the convergence study is demonstrated: assign different **Delta Error** and observe how the eigenmode solver converges to the theoretical result. The following figure displays the convergence study of the rectangle cavity with copper sidewalls:

MESHING										
level	freq(GHz)	nbTetr	Elapsed time	CPU time	nbUnknowns	mem(GB)	Elapsed time	CPU time	Delta(S)	
1	150	150	00:00:00.6	00:00:00.4	1296	0.043	00:00:00.6	00:00:01.5	/	
2	376	376	00:00:00.2	00:00:00.2	2892	0.067	00:00:01.2	00:00:03.4	0.002[->0.001]	
3	515	515	00:00:00.2	00:00:00.1	3904	0.066	00:00:01.7	00:00:04.1	4.5e-004[->0.001]	
SOLVING										
level	freq(GHz)	nbTetr	Elapsed time	CPU time	nbUnknowns	mem(GB)	Elapsed time	CPU time	Delta(S)	
1	150	150	00:00:00.7	00:00:00.4	1296	0.042	00:00:00.6	00:00:01.5	/	
2	376	376	00:00:00.2	00:00:00.2	2892	0.067	00:00:01.3	00:00:03.4	0.003[->0.000]	
3	515	515	00:00:00.2	00:00:00.2	3904	0.065	00:00:01.9	00:00:05.1	4.5e-004[->0.000]	
4	696	696	00:00:00.4	00:00:00.2	5188	0.077	00:00:02.3	00:00:06.3	1.9e-004[->0.000]	
5	901	901	00:00:00.3	00:00:00.2	6602	0.090	00:00:02.8	00:00:08.2	1.5e-004[->0.000]	
6	1189	1189	00:00:00.4	00:00:00.3	8544	0.106	00:00:03.9	00:00:11.0	3.8e-004[->0.000]	
7	1769	1769	00:00:00.5	00:00:00.4	12034	0.141	00:00:05.6	00:00:16.0	4.1e-005[->0.000]	

Mode		Fana	dF = 0.001	dF = 0.0001	Qana	dF = 0.001	dF = 0.0001
0	1 1	5.827E+09	5.826E+09	5.827E+09	10538.33	10523.28	10535.38
0	1 2	7.805E+09	7.806E+09	7.806E+09	13245.26	13175.77	13233.90
1	0 1	8.072E+09	8.075E+09	8.073E+09	10871.25	10804.88	10861.54
1	1 0	<b>9.008E+09</b>	<b>9.016E+09</b>	<b>9.009E+09</b>	<b>13086.41</b>	<b>12936.46</b>	<b>13055.47</b>
1	1 1	9.493E+09	9.495E+09	9.495E+09	N/A	9161.85	11382.99

With a smaller Delta error value (1e-4 instead of 1e-3), the mesh refinement takes four more passes to converge, but the results are more accurate compared to the theoretical results (Fana and Qana in table).