University of Nottingham

Netherlands Aerospace Centre

USER GUIDE

SACAMOS: State of the Art CAble MOdels for Spice
Open Souce Cable Models for EMI Simulations

AUTHORS
Christopher Smartt[1], David Thomas[1], Steve Greedy[1],
Jaco Verpoorte[2], Jesper Lansink Rotgerink[2] and Harmen Schippers[2]

CONTRACT:	ESA 40000112765/14/NL/HK.
DATE:	July 4, 2018
DOCUMENT VERSION:	1.2

[1] University of Nottingham, contact: Chris.Smartt@nottingham.ac.uk
[2] Netherlands Aerospace Centre, contact: Jaco.Verpoorte@nlr.nl

Contents

Chapter 1

Introduction

1.1 Software overview

The purpose of the SACAMOS software is to enable the creation of Spice cable models from the description/ characterisation of cables and bundles of cables together with information required to specify a particular modelling scenario (for example the specification of incident field excitation or transfer impedance coupling model). This User Guide relates to version 4.0.0 of the SACAMOS software.

A cable bundle may consist of multiple individual cables usually with the addition of a ground plane. This is illustrated in figure1.1 The cables may consist of shielded and/ or unshielded conductors.

The inputs to the tool consist of cable specifications, cable bundle specifications and specification of a simulation scenario. More specifically, a cable specification will consist of cable cross section geometry, materials and transfer impedance (if required). The bundle specification consists of the cable types and position in the bundle cross section, plus a ground plane specification (if required). The modelling scenario specification consists of the bundle length, transfer impedance coupling specification and incident field excitation as required.

The output is a cable model which takes the form of a Spice sub-circuit. The Spice model is dependent on the composition and configuration of the cable bundle, the bundle length and also the modelling scenario so a different model is required for each different incident field excitation (angle of incidence, polarisation) or crosstalk via transfer impedance study in which different source and victim conductors are identified.

The models are based on the Spice models described by Paul [1] and Xie [2] and further developed within this proejct at the University of Notingham. The theortical derivation of the Spice multi-conductor transmission line models is described in [3].

The software can produce Spice sub-circuit models of transmission lines for three versions of Spice:

1. Ngspice
2. LTspice

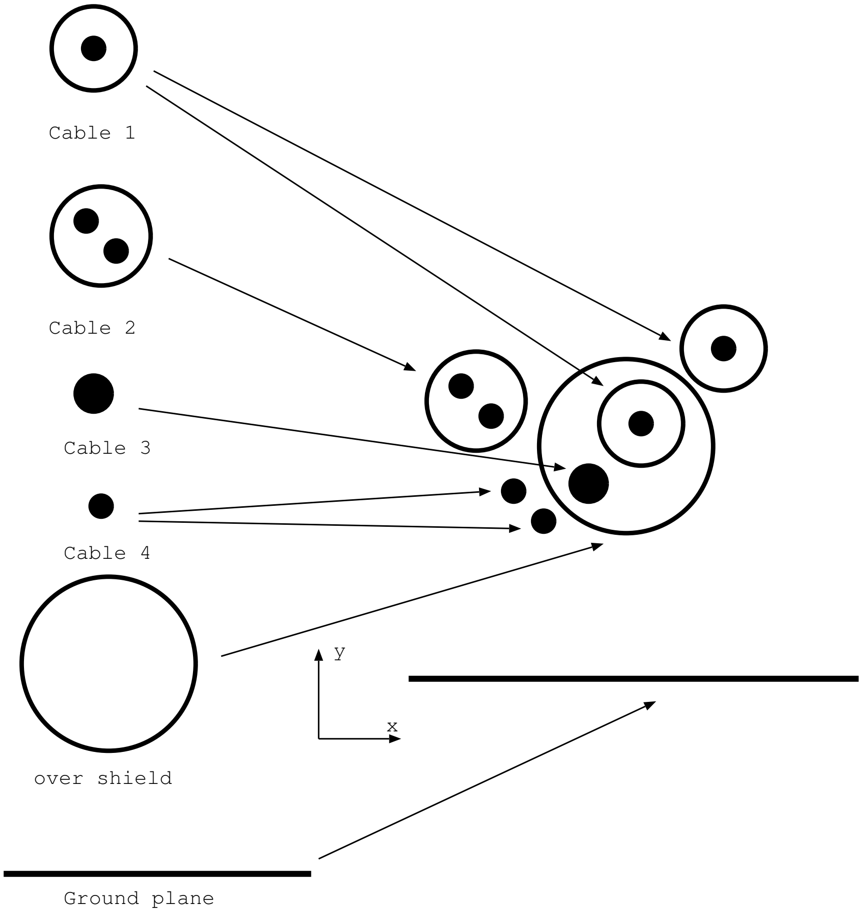


Figure 1.1 A complex cable bundle built up from individual cables, over-shield and ground plane

3. PSpice

1.1.1 Use of the Spice cable sub-circuit models

The Spice models of the transmission lines take the form of sub-cirucits which may be embedded in any circuit. The user should be aware of how the Spice transmission line sub-circuit models work with relation to the Spice node zero. Spice node zero is the global reference node and all voltages in the circuit are determined relative to this node.

If the transmission line is more than $1/10th$ of a wavelength then it is incorrect to use the same reference potential at both ends of the transmission line. This implies that care should be used in assigning node zero in a circuit containing transmission lines, especially if there is some loss (specifically d.c. resistance) associated with the reference conductor. In the case of d.c. resistance in the reference conductor, having node zero connected to the reference conductor at both ends shorts out this resistance which leads to inaccurate results, especially at low frequency.

The 'node zero' issue is discussed in more detail in section 9.3 however to summarise: in order to have accurate solutions the user should use the following guidelines:

If there is no loss in the reference conductor (e.g. a perfect ground plane) then it is permisible to have the reference conductor (ground) to be node zero at both ends of the transmission line sub-circuit.

If there is d.c. resistance associated with the reference conductor then the two ends of the reference conductor should have different nodes assigned to avoid short circuiting the d.c. resistance in the transmission line model.

Note that even if there are no common nodes at the two ends of the transmission line model then there is no problem with 'floating' sub-circuits.

1.2 Software modules

There are three processes in the creation of a Spice cable bundle model. These are:

1. cable model building process
2. cable bundle model building process
3. Spice cable bundle model building process

This decomposition of the process is suggested by the recognition that individual cable models can be re-used in different bundle specifications or used more than once in the same bundle specification. In addition, once a bundle has been specified, it may be used in different types of analysis for example a crosstalk assessment or the analysis of incident field excitation.

The cable models, bundle models and Spice cable bundle models may be kept within a library of cable models referred to in this project as MOD. This structure for the models is discussed in section 5.

Inputs and outputs to the processes consist of human readable text files thus the three stages of the model building process may be driven via a graphical user interface (GUI) or the processes may be run from the command line. Subsequent chapters describe the GUI and how to use it and also how to run the processes from the command line along with descriptions of the associated input and output file formats.

The software comes with a set of validation test cases which may be used to test the correct operation of the software. These test cases may also be used as examples and form the basis for the development of new cable bundle models. The test cases may be run automatically using a script which has been provided for the purpose. The running of the test cases is described in chapter 6.

1.2.1 Cable model building process

The cable model building process concerns the development of individual cable models from specifications. At this stage we can characterise the propagation within shielded conductor systems if they exist for a particular cable i.e. describe the parameters characterising wave propagation internal to any cable shields. In addition any transfer impedance models required to characterise the cable will be developed at this stage. The output of the cable model building process is a file containing a complete characterisation of the cable which may be used in any bundle configuration. The cable models can form the part of the library of cable models (MOD) relating to basic cable models.

1.2.2 Cable bundle model building process

The cable bundle model building process concerns the development of cable bundle models from individual cable models and additional ground and over-shield specifications. This stage involves the characterisation of the external propagation domain which involves the outside of the shields, unshielded conductors and ground plane and also the domain within over-shields.

1.2.3 Spice cable bundle model building process

Development of a Spice cable model for a particular bundle includes the specification of the length of the bundle, any incident field excitation and the specification of the coupling between cables via the transfer impedance of imperfect shields (i.e. identification of a shield and the direction of coupling to be included in the model for the chosen shield).

When run from the command line as opposed to the GUI, the Spice cable bundle model building process also allows the specification of a 'validation test case' in which the cable bundle model generated is exercised in a test case with simple resistor and series voltage source terminations on the conductors. Analytic and Spice model results (a.c. or transient transmission line teminatination voltage) are generated for the test configuration and may be compared directly.

1.3 Library of cable models (MOD)

The outputs of the three model building process stages will form elements of the library of cable models (MOD). The inputs to the cable bundle model builder and the Spice cable bundle model builder can also be drawn from MOD. The structure of MOD is described in chapter 5.

1.4 Structure of the User Guide

The document is structured as follows: Chapter 2 describes the creation of a cable model for each of the different cable types supported. Chapter 3 describes the creation of a cable bundle model. Chapter 4 describes the creation of a Spice cable bundle model. Chapter 6 describes the script which can be used to automatically run the test cases, compare validation results etc. Chapter 7 describes how to run the processes from the command line Chapter 5 decribes the structure of the library of cable models (MOD) which is generated by the test cases.

Chapter 2

Creating a Cable Model

2.1 Introduction

This chapter describes the creation of a cable model from specifications. The methods by which frequency dependent cable parameters (relative permittivity, finite conductivity loss models and transfer impedance models) are specified are described before detailing the specifications for each of the available cable types in turn. An example for each cable type is provided.

2.2 Cable Types

Models of the following cable types have been developed:

1. Cylindrical conductor with dielectric
2. Coaxial cable with transfer impedance and shield surface impedance loss
3. Twinax cable with transfer impedance and shield surface impedance loss
4. Twisted pair
5. Shielded twisted pair with transfer impedance and shield surface impedance loss
6. Spacewire cable with transfer impedance and shield surface impedance loss
7. Overshield with transfer impedance and shield surface impedance loss
8. flex cable
9. D connector

All of the cable types with the exception of the D connector allow the inclusion of frequency dependent dielectrics and finite conductivity models of conductors.

The typical values required to specify each of these cable types is described in the following sub-sections. Each cable type is illustrated in a figure. The figure shows the conductor numbering used in the software e.g. conductor number 1 of a coaxial cable is the inner conductor and conductor number 2 is the shield.

In addition to these cable types a perfectly conducting ground plane is available when building a cable bundle.

2.2.1 Frequency dependent models

Many of the cable models available have frequency dependent properties. The frequency dependent properties arise from frequency dependent permittivity of dielectrics, from the finite conductivity of conductors and from the frequency dependence of transfer impedance. The frequency dependent models are described in detail in the Theory Manual [3] sections 3.6 and chapter 5.

Frequency dependent rational function dielectric models

Frequency dependent cable properties (dielectric relative permittivity or transfer impedance) are defined as rational functions of frequency. The rational function includes a normalisation constant ω_0 which is used to prevent the coefficients of the function becoming too large or too small i.e.

$$\epsilon_r\left(j\omega\right) = \frac{a_0 + a_1\left(\frac{j\omega}{\omega_0}\right) + a_2\left(\frac{j\omega}{\omega_0}\right)^2 + \dots}{b_0 + b_1\left(\frac{j\omega}{\omega_0}\right) + b_2\left(\frac{j\omega}{\omega_0}\right)^2 + \dots} \tag{2.1}$$

For example, a Debye dielectric model has a relative permittivity described by

$$\epsilon_r = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1 + j\omega\tau} \tag{2.2}$$

where ϵ_∞ is the relative permittivity at the high frequency limit, ϵ_s is the relative permittivity at the low frequency limit and τ is the relaxation time of the material.

this may be written in the form of equation2.1 as

$$\epsilon_r\left(j\omega\right) = \frac{\epsilon_s + \epsilon_\infty\left(\frac{j\omega}{\omega_0}\right)}{1 + \left(\frac{j\omega}{\omega_0}\right)} \tag{2.3}$$

where

$$\omega_0 = \frac{1}{\tau} \tag{2.4}$$

The frequency dependence of relative permittivity cannot be aritrarily specified as the real and imaginary parts of the realtive permittivity are related through the Kramers-Kronig relations [4]. This relationship is due to the causality of the dielectric time response and is fundamental to linear systems. The implications for modelling dielectrics is that the permittivity model used must satisfy the Kramers-Kronig relations.

The rational function representation of the relative permittivity in equation 2.1 has the advantage of naturally satisfying these relations. If the real part of the permittivity is a function of frequency then the Kramer Kronig relations imply that this will be associated with some loss and vice-versa. We note that the common approximation in which the imaginary part of permittivity is assumed to be constant (constant $tan\,(\delta)$ model) is unphysical.

It is difficult to obtain the good quality complex relative permittivity data over a wide frequency band required to generate frequency dependent dielectric models, even for the dielectric materials used in the construction of cables. Unless very good data is available we recommend that a simple constant relative permittivity model be used.

If tabulated frequency domain relative permittivity data is available then a model fitting process may be applied to generate the rational function coefficients for a frequency dependent dielectric model. This process is described in section 7.4.

Frequency dependent finite conductivity loss models

Cable losses arising from the finite conductivity of a conductor are incorporated into some of the cable models available [3]. In these models the conductivity of the conductors are specified as parameters of the cable model. The contribution to the impedance terms due to the finite conductivity is the surface impedance of the conductor [5]. The surface impedance is frequency dependent and incorporates the 'skin effect' into the model. For cylindrical conductors analytic expressions are available for the surface impedance of solid cylindrical conductors and cylindrical shells (cable shields).

For a solid cylindrical conductor of radius, r, permeability, μ and conductivity, σ the internal impedance due to the magnetic field penetrating the conductor at frequency f is given by

$$Z_{int\_cylinder} = \frac{1}{\sqrt{2}\pi r\sigma\delta}\left(\frac{ber(q)+jbei(q)}{bei'(q)-jber'(q)}\right) \tag{2.5}$$

where ber and bei are Kelvin functions, δ is the skin depth given by

$$\delta = \frac{1}{\sqrt{\pi f\mu\sigma}} \tag{2.6}$$

and q is

$$q = \sqrt{2}\frac{r}{\delta} \tag{2.7}$$

For a cylindrical shell i.e. a cable shield, the surface impedance (neglecting small terms related to the curvature of the conductor) may be evaluated as follows:

The d.c. resistance of a shell of radius, r, and thickness, t, is

$$R_{dc} = \frac{1}{\sqrt{2}\pi\sigma rt} \tag{2.8}$$

The complex propagation constant in the conductor is

$$\gamma = \frac{(1+j)}{\delta} \tag{2.9}$$

Then the surface impedance of the cylindrical shell is

$$Z_{int\_shell} = R_{dc}\gamma t\; cosech\left(\gamma t\right) \tag{2.10}$$

It is important to note that at low frequency the surface impedance is equal to the d.c. resistance of the shield and that the transfer impedance should also take this value.

Note that if a transfer impedance model is included then the shield thickness parameter can be set to zero in which case the software will calculate an ‘equivalent thickness’ which gives the correct d.c. resistance for the shield, based on the d.c. transfer resistance and the specified conductivity of the shield.

The loss model for rectangular conductors assumes that the internal impedance of the conductor takes the form

$$Z_{int\_rectangular} = R_{dc} + B\sqrt{j\omega} \tag{2.11}$$

where R_{dc} is the d.c. resistance of a rectangular wire of width w, and height, t, is given by

$$R_{dc} = \frac{1}{\sigma w t} \tag{2.12}$$

and B is given by

$$B = \frac{1}{2\left(w+t\right)}\sqrt{\frac{\mu}{\sigma}} \tag{2.13}$$

These internal impedances then contribute to the impedance matrix of the cable as appropriate for each configuration.

Frequency dependent transfer impedance model

Some shielded cable models allow the impedance of the shield conductor (transfer impedance) to be specified as a frequency dependent function [3]. Cable models with frequency dependent transfer impedance models are:

1. Coaxial cable with transfer impedance and shield surface impedance loss
2. Twinax cable with transfer impedance and shield surface impedance loss
3. Shielded twisted pair with transfer impedance and shield surface impedance loss
4. Spacewire cable with transfer impedance and shield surface impedance loss
5. Overshield with transfer impedance and shield surface impedance loss

The frequency dependent transfer impedance is represented using a rational function form i.e.

$$Z_T\left(j\omega\right) = \frac{a_0 + a_1\left(\frac{j\omega}{\omega_0}\right) + a_2\left(\frac{j\omega}{\omega_0}\right)^2 + \ldots}{b_0 + b_1\left(\frac{j\omega}{\omega_0}\right) + b_2\left(\frac{j\omega}{\omega_0}\right)^2 + \ldots} \tag{2.14}$$

Hence the cable specification requires the coefficients of the rational function, along with the frequency scaling, ω_0 to be specified. This model has the often used $Z_T =$

$R_T + j\omega L_T$ model as a special case where $a_0 = R_T$, $a_1 = L_T$, $\omega_0 = 1.0$ and $b_0 = 1.0$. The coefficients for more complex transfer impedance models may be generated from the specifications of a braided shield as in reference [7] followed by a rational function fitting to the resulting frequency domain transfer impedance function. This process is described in section 7.5. Alternatively the rational function coefficiants may be calculated by rational function fitting to measured (or otherwise obtained) transfer imepdance data as described in section 7.4. The transfer impedance of a shield is reciprocal thus the coupling through the shield is determined by the same transfer impedance for both coupling directions.

It is important to note that for a model to be self consistent at low frequency the transfer impedance of a shield should be equal to the d.c. resistance of the shield and hence also the low frequency surface impedance i.e.

$$Z_T\left(\omega = 0\right) = \frac{a_0}{b_0} = Z_{int\_shell} = R_{dc} \tag{2.15}$$

2.2.2 Cable Specification File Format

This section describes the cable specification file format used as the input to the cable model building process. Cable specification files have the extension **name.cable_spec**.

The first lines of the **.cable_spec** file specifies the path to the cable model to be produced from the specification (i.e. the path to the cable models within MOD). Following this come the parameters which define the cable i.e. the geometrical description of the cable cross section, dielectric properties (if required) and transfer impedance specification (if required).

In addition to the data required to specify a cable bundle, additional information and flags may be specified to influence the operation of the software. There is a choice whether to use (approximate) analytic formulae to calculate the per-unit-length parameters of shielded domains within a cable or to use the numerical Laplace solver.

The approximate analytic formula for shielded domains uses a ‘wide separation’ approximation i.e. it is assumed that the conductor radii are small compared to their separation.

If frequency dependent dielectric models are used in the cable specification then the filter fitting process will be required to set the elements of the cable admittance matrices for shielded domains. This is discussed in detail in the Frequency Dependent Transfer Functions chapter of the theory manual [3]. The parameters of the filter fitting process may be set following the cable specification. The filter fitting process provides a best fit model of specified order over a specified frequency range. As a default the model order is 0 i.e. no frequency dependence is included in the admittance matrix (the permittivity used is the high frequency value of the dielectric constant). The model order can be specified in two ways:

1. The order is specified as a positive integer and this is the order used
2. A negative integer is specified. In this case the order is chosen using an automatic algorithm which attempts to choose the best order from 0 up to |specified order|

Following the model order the frequency range is specified. The frequency scale is set to be either linear (‘lin’) or logarithmic (‘log’), following this the minimum frequency, maximum frequency and the number of frequencies for the filter fitting process are specified. If the Laplace solution is used then the number of frequencies should not be too large as this will lead to excessive runtimes for the cable model building process.

An example of a filter fitting specification for the admittance matrix element fit is shown below in which the best model up to order 10 is chosen based on a fit to logarithmic frequency data over a range of 10kHz to 1GHz with 16 sample points:

```
-10          # order for admittance matrix element fit model
log          # frequency scale  (log or lin)
1e5 1e9 16   # fmin fmax number_of_frequencies
```

The flags which may be applied in a **cable_spec** file are as follows:

1. ‘verbose’ output detailed summary of the software operation and calculation results.

2. 'use_Laplace' use the numerical Laplace solver to calculate inductance and capacitance matrices for the internal domains. By default, approximate analytic formulae are used.

3. 'no_Laplace' use the (approximate) analytic formulae to calculate inductance and capacitance matrices for the internal domains.

4. 'plot_mesh' output a vtk file which shows the mesh used in Finite Element Laplace calculations.

5. 'direct_solver' Solve the Finite Element matrix equation using a direct solver i.e. by calculating the matrix inverse.

6. 'iterative_solver' Solve the Finite Element matrix equation using an iterative solver. This is much more efficient than the direct solver and is the default option

If the Laplace solver is used then the mesh generation is controlled by the parameter 'Laplace_surface_mesh_constant' This parameter determines the number of finite element edges on a conductor surface. The edge length of elements on a cylindrical conductor of radius r is $\frac{r}{Laplace\_surface\_mesh\_constant}$. The default value is 3.

The default value is a compromise between accuracy and computation time for the Laplace solution. The default value may be overridden by the user by appending the following to the end of the **.cable_spec** file:

```
Laplace_surface_mesh_constant
5
```

If the iterative solver is used to solve the Finite Element matrix equations then the tolerance for the iterative solution may be set using the parameter cg_tol. The default value is 10^{-12}. The parameter may be set (in this example to 1E-14) by including the following lines int the **.cable_spec** file:

```
cg_tol
1E-14
```

2.2.3 Cable models available

The available cable models are described below. For each cable type a figure is provided which shows the cable cross section and the conductor numbering used for the cable. The parameters required to specify a cable are outlined and an example **.cable_spec** file is provided.

The cable models are very general in that all shields can have a transfer impedance specified, all dielectrics can be frequency dependent and all conductors can have a finite conductivity specified. It may often be the case that not all the information required to specifiy a cable is available or only a simple model is required. If that is the case then the general model may be simplified in the following ways:

1. No dielectric on the outside of a cable: This may be included by setting the relative permittivity of the dielectric to 1 or by setting the dielectric radius equal to the conductor radius.
2. Lossless i.e. perfect conductors. This can be incuded in the model by setting the conductivity parameter to zero. This indicates to the software that loss is not to be included in the model.
3. No transfer impedance to be included. A transfer impedance must be specified however it can be set to zero.

As was stated in section 2.2.1, the transfer impedance of a shield at d.c. should be equal to the d.c. resistance of the shield. In order to avoid the need for the user to perform any calculations to ensure that this is the case the thickness parameter for a shield can be set to zero in the **.cable_spec** file. When the shield thickness parameter is set to zero the software calculates an 'equivalent shield thickness' based on the d.c. value of the transfer impedance (R_{dc}) and the specified conductivity of the shield i.e.

$$t = \frac{1}{2\pi r_s \sigma R_{dc}} \tag{2.16}$$

Frequency dependent cylindrical conductor with dielectric

Figure 2.1 shows the cross section of the cylindrical cable model with dielectric. A description of the cable parameters is given in table 2.1 followed by an example.

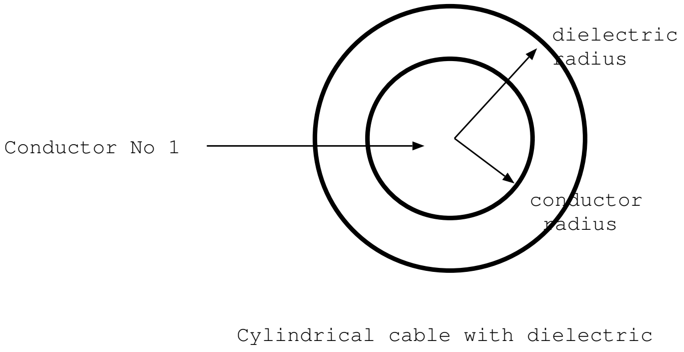


Figure 2.1 Cylindrical conductor with dielectric

example value	unit	Comment
1	integer	Number of conductors
3	integer	Number of parameters
0.25e-3	metre	parameter 1: conductor radius
0.5e-3	metre	parameter 2: dielectric radius
5e7	Siemens/metre	parameter 3: electric conductivity
1	integer	number of frequency dependent parameters
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent dielectric relative permittivity model

Table 2.1 Cylindrical cable parameters

Example

```
#MOD_cable_lib_dir
.
Cylindrical
1 # number of conductors
3 # number of parameters
1.905e-4   # parameter 1: conductor radius
0.5e-3   # parameter 2: dielectric radius
5E7           # parameter 3: conductivity
1 # number of frequency dependent parameters
# dielectric relative permittivity model follows
   1E8          # w normalisation constant
           1    # a order, a coefficients follow below:
  2.60        2.25
           1    # b order, b coefficients follow below:
  1.0         1.0
```

Frequency Dependent Coaxial cable with Transfer Impedance and surface impedance loss

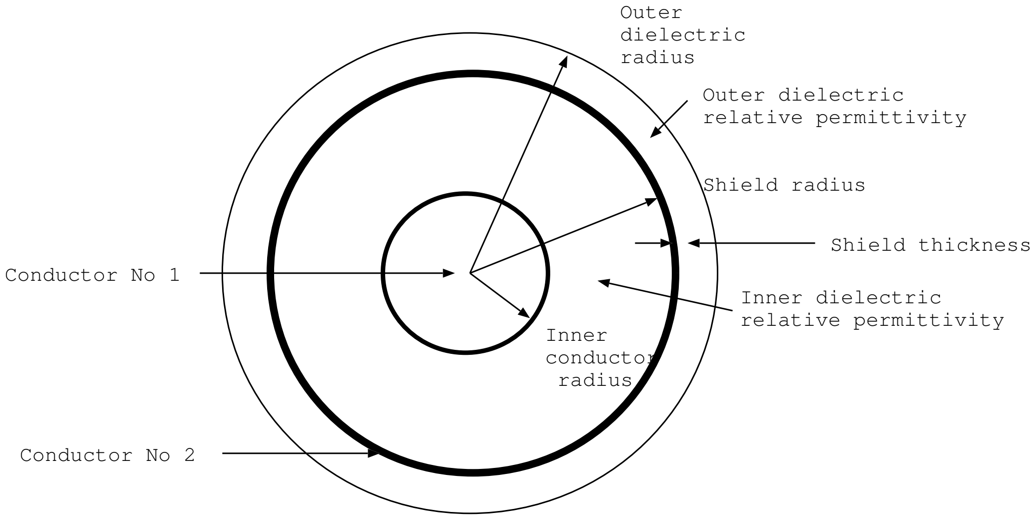
Figure 2.2 shows the cross section of the frequency dependent coaxial cable with transfer impedance and surface impedance loss. A description of the cable parameters is given in table 2.2 followed by an example.

The inductance and capacitance of the coaxial mode is always calculated using the analytic formulae [1]

$$L = \frac{\mu_0}{2\pi} \ln\left(\frac{r_s}{r_w}\right) \tag{2.17}$$

$$C = \frac{2\pi\epsilon_0\epsilon_r\left(j\omega\right))}{\ln\left(\frac{r_s}{r_w}\right)} \tag{2.18}$$

If the dielectric is frequency dependent then the frequency dependent capacitance is simply a scaling of the dielectric frequency dependent function specified.



Coaxial cable with surface impedance loss

Figure 2.2 Coaxial cable

example value	unit	Comment
2	integer	Number of conductors
6	integer	Number of parameters
0.25e-3	metre	parameter 1: inner conductor radius
1.25e-3	metre	parameter 2: shield radius
2.5e-3	metre	parameter 3: outer dielectric radius
5e7	Siemens/metre	parameter 4: inner conductor electric conductivity
0.05E-3	metre	parameter 5: shield conductor thickness
5e7	Siemens/metre	parameter 6: shield conductor electric conductivity
2	integer	number of frequency dependent parameters
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent inner dielectric relative permittivity model
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent outer dielectric relative permittivity model
1	integer	number of frequency dependent transfer impedance models
$Z_T\left(j\omega\right)$	ohms (rational function coefficients)	Frequency dependent transfer impedance model

Table 2.2 Coaxial cable parameters

Example

```
#MOD_cable_lib_dir
LIBRARY_OF_CABLE_MODELS
Coax
2 # number of conductors
6 # NUMBER OF PARAMETERS
0.00042   # parameter 1: inner conductor radius (m)
0.00147  # parameter 2: shield radius (m)
0.0025          # parameter 3: outer insulation radius (m)
5e7             # parameter 4: inner conductor electric conductivity
0.0002          # parameter 5: shield conductor thickness
5e7             # parameter 6: shield electric conductivity
2               # number of frequency dependent parameters
# Inner dielectric relative permittivity model follows
   1E7          # w normalisation constant
           1    # a order, a coefficients follow below:
  2.6        2.0
           1    # b order, b coefficients follow below:
  1.0         1.0
```

```
# Outer dielectric relative permittivity model follows
   1E7          # w normalisation constant
           1    # a order, a coefficients follow below:
  2.40        2.20
           1    # b order, b coefficients follow below:
  1.0         1.0
1               # number of frequency dependent transfer ipedance models
# Transfer impedance model
1.0             # angular frequency normalisation
1               # order of numerator model
0.05  1.6E-9    # list of numerator coefficients a0 a1 a2...
0               # order of denominator model
1.0             # list of denominator coefficients b0 b1 b2...
```

Frequency Dependent Twinaxial cable with transfer impedance with shield surface impedance loss

Figure 2.3 shows the cross section of the frequency dependent twinax cable with transfer impedance and surface impedance loss. A description of the cable parameters is given in table 2.3 followed by an example in which the filter fitting model order is specified and the Laplace solver used for the frequency dependent per-unit length parameter calculation.

If the approximate analytic solution is used to calculate the per-unit-length parameters of the internal modes then the solution assumes that the space within the shield is completely filled with the inner dielectric. If the inner dielectric is frequency dependent then the frequency dependence is neglected and the high frequency permittivity value is used. The analytic solution uses the wide separation approximation [1].

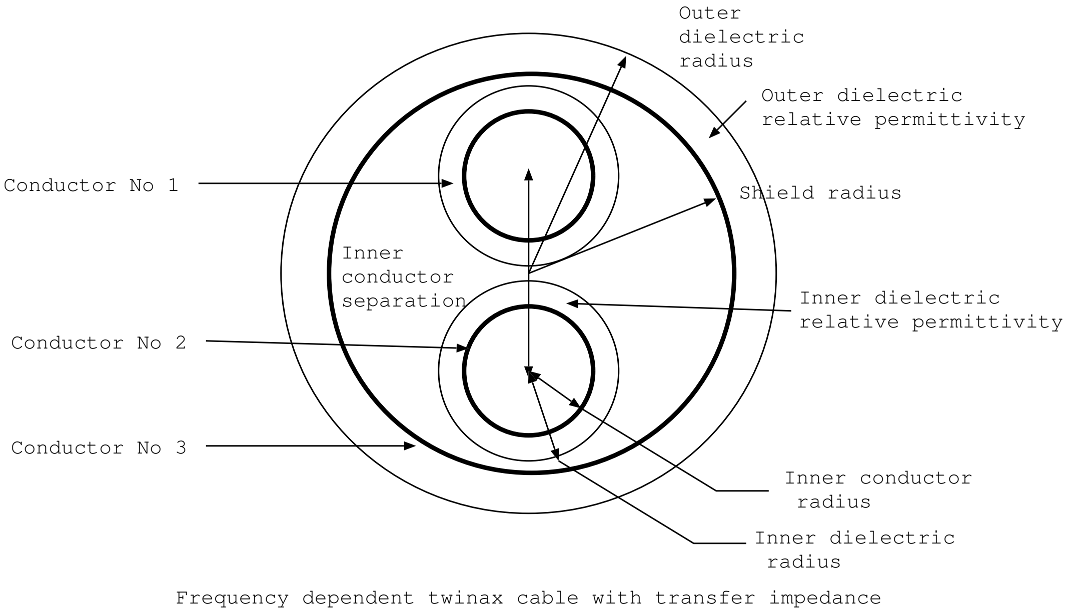

Outer dielectric radius
Outer dielectric relative permittivity
Conductor No 1
Shield radius
Inner conductor separation
Inner dielectric relative permittivity
Conductor No 2
Conductor No 3
Inner conductor radius
Inner dielectric radius
Frequency dependent twinax cable with transfer impedance

Figure 2.3 Twinaxial cable

example value	unit	Comment
3	integer	Number of conductors
8	integer	Number of parameters
0.25e-3	metre	parameter 1: inner conductor radius
0.40e-3	metre	parameter 2: inner dielectric radius
1.0e-3	metre	parameter 3: inner conductor separation
2.0e-3	metre	parameter 4: shield radius
0.1e-3	metre	parameter 5: shield thickness
2.5e-3	metre	parameter 6: outer dielectric radius
5e7	Siemens/metre	parameter 7: inner conductor electric conductivity
5e7	Siemens/metre	parameter 8: shield electric conductivity
2	integer	number of frequency dependent parameters
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent inner dielectric relative permittivity model
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent outer dielectric relative permittivity model
1	integer	number of frequency dependent transfer impedance models
$Z_T\left(j\omega\right)$	ohms (rational function coefficients)	Frequency dependent transfer impedance model

Table 2.3 Twinaxial cable parameters

Example

```
#MOD_cable_lib_dir
.
Twinax
3               # Number of conductors
8               # Number of parameters
0.25e-3         # parameter 1: inner conductor radius
0.40e-3         # parameter 2: inner dielectric radius
1.0e-3          # parameter 3: inner conductor separation
2.0e-3          # parameter 4: shield radius
0.1e-3          # parameter 5: shield thickness
2.5e-3          # parameter 6: outer dielectric radius
5e7             # parameter 7: inner conductor electric conductivity
5e7             # parameter 8: shield electric conductivity
2         # number of frequency dependent parameters
# Inner dielectric relative permittivity model follows
   1E7          # w normalisation constant
```

```
           1   # a order, a coefficients follow below:
  2.6        2.0
           1   # b order, b coefficients follow below:
  1.0         1.0
# Outer dielectric relative permittivity model follows
   1E7         # w normalisation constant
           1   # a order, a coefficients follow below:
  2.40        2.20
           1   # b order, b coefficients follow below:
  1.0         1.0
1              # number of frequency dependent transfer ipedance models
# Transfer impedance model
1.0            # angular frequency normalisation
1              # order of numerator model
0.05  1.6E-9   # list of numerator coefficients a0 a1 a2...
0              # order of denominator model
1.0            # list of denominator coefficients b0 b1 b2...
-10    ! order for filter fit solution
lin   # frequency range type for filter fitting type (lin or dB)
1e3 1e9 10 # fmin fmax number_of_frequencies for filter fitting
use_Laplace
```

Frequency Dependent Twisted pair cable

Figure 2.4 shows the cross section of the frequency dependent twisted pair model. A description of the cable parameters is given in table 2.4 followed by an example. Note that there is no parameter for the twisted pair model which relates to the number of twists per unit length. This is due to the nature of the model which assumes that there is no interaction between the differential mode and the other conductors in the bundle thus the twisting period is not required for the model.

The analytic formulae used to calculate the inductance and capacitance of the differential mode are [1]

$$C = \frac{\pi\epsilon_0}{\ln\left(\frac{s}{2r_w} + \sqrt{\left(\frac{s}{2r_w}\right)^2 - 1}\right)} \tag{2.19}$$

$$L = \frac{\mu_0\epsilon_0}{C} \tag{2.20}$$

where the dielectric coating of the conductors is neglected.

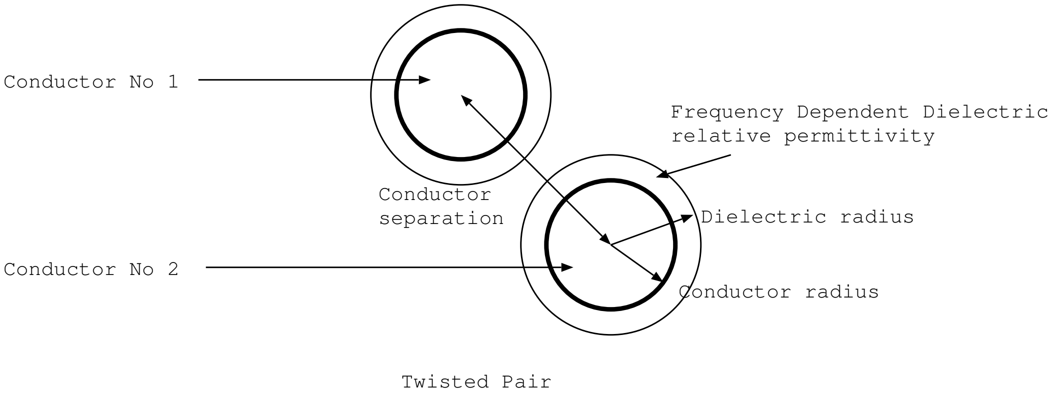


Figure 2.4 Twisted pair cable

example value	unit	Comment
2	integer	Number of conductors
4	integer	Number of parameters
0.25e-3	metre	parameter 1: conductor radius, r_w
1.0e-3	metre	parameter 2: conductor separation, s
0.5e-3	metre	parameter 3: dielectric radius
5e7	Siemens/metre	parameter 4: inner conductor electric conductivity
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent dielectric relative permittivity model

Table 2.4 Twisted pair cable parameters

Example

```
#MOD_cable_lib_dir
.
Twisted_pair
2 # number of conductors
4 # number of parameters
0.25e-3   # parameter 1: conductor radius
1.0e-3   # parameter 2: conductor separation
0.45e-3   # parameter 3: dielectric radius
5e7             # parameter 4: inner conductor electric conductivity
# Dielectric relative permittivity model follows
   1E7          # w normalisation constant
           1    # a order, a coefficients follow below:
  2.40        2.20
           1    # b order, b coefficients follow below:
  1.0         1.0
```

Frequency Dependent shielded twisted pair cable with transfer impedance and shield surface impedance loss

Figure 2.5 shows the cross section of the frequency dependent shielded twisted pair cable with transfer impedance and shield surface impedance loss model. A description of the cable parameters is given in table 2.5 followed by an example in which the filter fitting model order is specified and the Laplace solver used for the frequency dependent per-unit length parameter calculation. Note that there is no parameter for the number of twists per unit length. This is due to the nature of the model which assumes that there is no interaction between the differential mode and the other conductors in the bundle thus the twisting period is not required for the model.

If the approximate analytic solution is used to calculate the per-unit-length parameters of the internal modes then the solution assumes that the space within the shield is completely filled with the inner dielectric. If the inner dielectric is frequency dependent then the frequency dependence is neglected and the high frequency permittivity value is used. The analytic solution uses the wide separation approximation [1] .

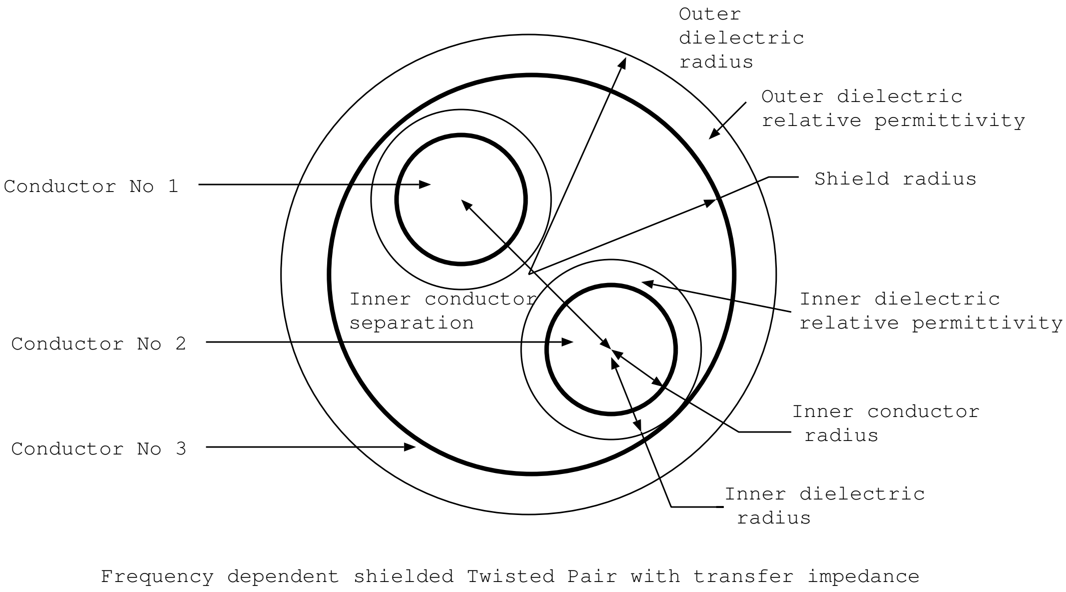

Outer dielectric radius
Outer dielectric relative permittivity
Shield radius
Conductor No 1
Inner conductor separation
Inner dielectric relative permittivity
Conductor No 2
Inner conductor radius
Conductor No 3
Inner dielectric radius
Frequency dependent shielded Twisted Pair with transfer impedance

Figure 2.5 Shielded twisted pair

example value	unit	Comment
3	integer	Number of conductors
8	integer	Number of parameters
0.25e-3	metre	parameter 1: inner conductor radius
0.40e-3	metre	parameter 2: inner dielectric radius
1.0e-3	metre	parameter 3: inner conductor separation
2.0e-3	metre	parameter 4: shield radius
0.1e-3	metre	parameter 5: shield thickness
2.5e-3	metre	parameter 6: outer dielectric radius
5e7	Siemens/metre	parameter 7: inner conductor electric conductivity
5e7	Siemens/metre	parameter 8: shield electric conductivity
2	integer	number of frequency dependent parameters
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent inner dielectric relative permittivity model
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent outer dielectric relative permittivity model
1	integer	number of frequency dependent transfer impedance models
$Z_T\left(j\omega\right)$	ohms (rational function coefficients)	Frequency dependent transfer impedance model

Table 2.5 Shielded twisted pair parameters

Example

```
#MOD_cable_lib_dir
.
Shielded_twisted_pair
3             # Number of conductors
8             # Number of parameters
0.25e-3       # parameter 1: inner conductor radius
0.40e-3       # parameter 2: inner dielectric radius
1.0e-3        # parameter 3: inner conductor separation
2.0e-3        # parameter 4: shield radius
0.1e-3        # parameter 5: shield thickness
2.5e-3        # parameter 6: outer dielectric radius
5e7           # parameter 7: inner conductor electric conductivity
5e7           # parameter 8: shield electric conductivity
2       # number of frequency dependent parameters
# Inner dielectric relative permittivity model follows
   1E7        # w normalisation constant
```

```
          1  # a order, a coefficients follow below:
  2.6       2.0
          1  # b order, b coefficients follow below:
  1.0        1.0
# Outer dielectric relative permittivity model follows
   1E7       # w normalisation constant
          1  # a order, a coefficients follow below:
  2.40       2.20
          1  # b order, b coefficients follow below:
  1.0        1.0
1            # number of frequency dependent transfer ipedance models
# Transfer impedance model
1.0          # angular frequency normalisation
1            # order of numerator model
0.05  1.6E-9  # list of numerator coefficients a0 a1 a2...
0            # order of denominator model
1.0          # list of denominator coefficients b0 b1 b2...
-10    ! order for filter fit solution
lin   # frequency range type for filter fitting type (lin or dB)
1e3 1e9 10 # fmin fmax number_of_frequencies for filter fitting
use_Laplace
```

Frequency dependent Spacewire cable with transfer impedance and shield surface impedance loss

Figure 2.6 shows the cross section of the frequency dependent spacewire cable with transfer impedance and shield surface impedance loss model. A description of the cable parameters is given in table 2.6 followed by an example in which the filter fitting model order is specified and the Laplace solver used for the frequency dependent per-unit length parameter calculation. Note that there is no parameter for the number of twists per unit length in the shielded twisted pairs. This is due to the nature of the model which assumes that there is no interaction between the differential mode and the other conductors in the bundle thus the twisting period is not required for the model.

If the approximate analytic solution is used to calculate the per-unit-length parameters of the internal modes then the solution assumes that the space within the twisted pair shields is completely filled with the inner dielectric. If the inner dielectric is frequency dependent then the frequency dependence is neglected and the high frequency permittivity value is used. Similarly the solution assumes that the space within the outer shield is completely filled with the inner shield dielectric. If the inner shield dielectric is frequency dependent then the frequency dependence is neglected and the high frequency permittivity value is used. The analytic solution uses the wide separation approximation [1].

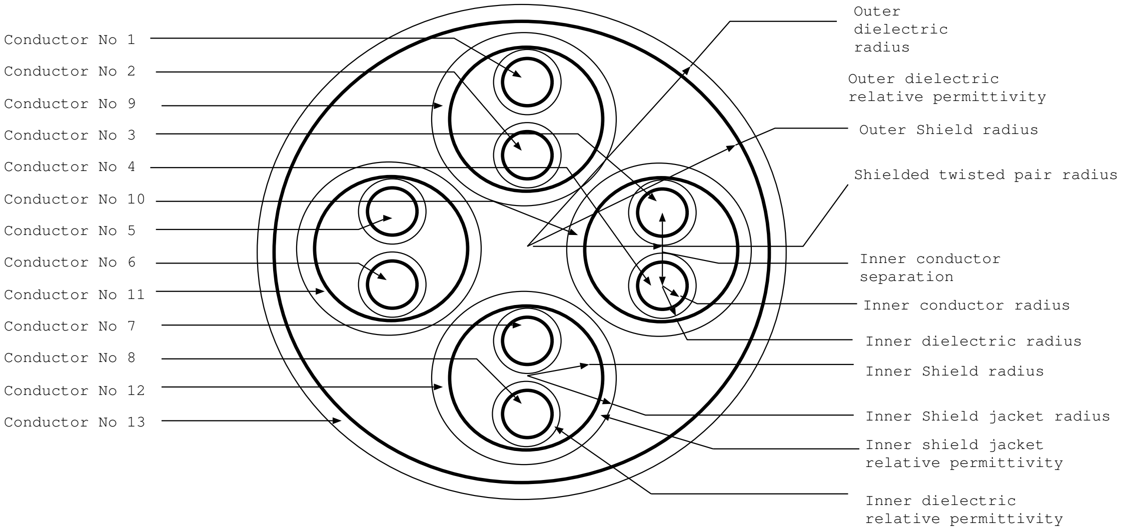


Figure 2.6 Spacewire

example value	unit	Comment
13	integer	Number of conductors
13	integer	Number of parameters
0.25e-3	metre	parameter 1: inner conductor radius
0.40e-3	metre	parameter 2: inner dielectric radius
1.0e-3	metre	parameter 3: inner conductor separation
2.0e-3	metre	parameter 4: inner shield radius
0.1e-3	metre	parameter 5: inner shield thickness
2.25e-3	metre	parameter 6: inner shield jacket radius
3.25e-3	metre	parameter 7: shielded twisted pair radius
5.65e-3	metre	parameter 8: outer shield radius
0.15e-3	metre	parameter 9: outer shield thickness
6.25e-3	metre	parameter 10: outer dielectric radius
5e7	Siemens/metre	parameter 11: inner conductor electric conductivity
5e7	Siemens/metre	parameter 12: inner shield electric conductivity
5e7	Siemens/metre	parameter 13: outer shield electric conductivity
3	integer	number of frequency dependent parameters
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent inner dielectric relative permittivity model
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent inner shield dielectric relative permittivity model
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent outer dielectric relative permittivity model
2	integer	number of frequency dependent transfer impedance models
$Z_T\left(j\omega\right)$	ohms (rational function coefficients)	Inner shield Frequency dependent transfer impedance model
$Z_T\left(j\omega\right)$	ohms (rational function coefficients)	Outer shield Frequency dependent transfer impedance model

Table 2.6 Spacewire parameters

Example

```
#MOD_cable_lib_dir
.
Spacewire
13 # number of conductors
13              # Number of parameters
0.25e-3         # parameter 1: inner conductor radius
0.40e-3         # parameter 2: inner dielectric radius
1.0e-3          # parameter 3: inner conductor separation
2.0e-3          # parameter 4: inner shield radius
0.1e-3          # parameter 5: inner shield thickness
2.25e-3         # parameter 6: inner shield jacket radius
3.25e-3         # parameter 7: shielded twisted pair radius
5.65e-3          # parameter 8: outer shield radius
0.1e-3          # parameter 9: outer shield thickness
6.25e-3         # parameter 10: outer dielectric radius
5e7             # parameter 11: inner conductor electric conductivity
5e7             # parameter 12: inner shield electric conductivity
5e7             # parameter 13: outer shield electric conductivity
3        number of frequency dependent parameters
# Inner dielectric relative permittivity model follows
   1E7          # w normalisation constant
           1    # a order, a coefficients follow below:
  2.6        2.0
           1    # b order, b coefficients follow below:
  1.0         1.0
# Inner shield dielectric relative permittivity model follows
   1E7          # w normalisation constant
           1    # a order, a coefficients follow below:
  2.6        2.0
           1    # b order, b coefficients follow below:
  1.0         1.0
# Outer dielectric relative permittivity model follows
   1E7          # w normalisation constant
           1    # a order, a coefficients follow below:
  2.40        2.20
           1    # b order, b coefficients follow below:
  1.0         1.0
2               # number of frequency dependent transfer ipedance models
# Inner shield Transfer impedance model
1.0             # angular frequency normalisation
1               # order of numerator model
0.05  1.6E-9    # list of numerator coefficients a0 a1 a2...
0               # order of denominator model
1.0             # list of denominator coefficients b0 b1 b2...
```

```
# Outer shield Transfer impedance model
1.0             # angular frequency normalisation
1               # order of numerator model
0.002  2.8E-9   # list of numerator coefficients a0 a1 a2...
0               # order of denominator model
1.0             # list of denominator coefficients b0 b1 b2...
-10    ! order for filter fit solution
lin   # frequency range type for filter fitting type (lin or dB)
1e3 1e9 10 # fmin fmax number_of_frequencies for filter fitting
use_Laplace
```

Over-Shield

Figure 2.7 shows the cross section of the overshield. A description of the overshield parameters is described in table 2.7 followed by an example.

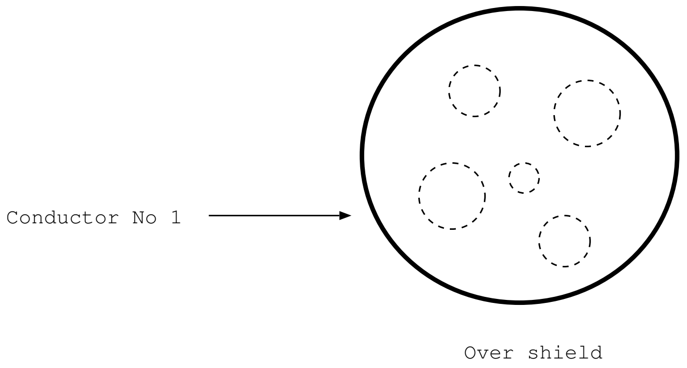


Figure 2.7 Over shield

example value	unit	Comment
1	integer	Number of conductors
3	integer	Number of parameters
5.5e-3	metre	parameter 1: overshield radius
0.05E-3	metre	parameter 2: overshield conductor thickness
5e7	Siemens/metre	parameter 3: overshield conductor electric conductivity
0	integer	number of frequency dependent parameters
1	integer	number of frequency dependent transfer impedance models
$Z_T\left(j\omega\right)$	ohms (rational function coefficients)	overshield Frequency dependent transfer impedance model

Table 2.7 Overshield parameters

Example

```
#MOD_cable_lib_dir
LIBRARY_OF_CABLE_MODELS
Overshield
1        # number of conductors
3        # number of parameters
0.005        # parameter 1: overshield radius
0.0001        # parameter 2: overshield thickness
5E7        # parameter 3: overshield conductivity
0              # number of frequency dependent parameters
1              # number of frequency dependent transfer impedance models
# Transfer impedance model
1.0            #  angular frequency normalisation
1              #  order of numerator model
0.05  1.6E-9   #  list of numerator coefficients a0 a1 a2...
0              #  order of denominator model
1.0            #  list of denominator coefficients b0 b1 b2...
```

Flex cable

Version 4.0.0 of the SACAMOS software introduces a new flex cable model in which more than one row of conductors can be included within the rectangular dielelctric region. This allows flex cables with multiple layers as well as cables with different conductor dimensions and spacings to be simulated. Figure 2.8 shows the cross section of the new flex cable model. A description of the model parameters is described in table 2.8 followed by an example. The flex cable differs from most other cable models in that the number of conductors is a parameter of the model. The conductors are numbered from left to right in each row.

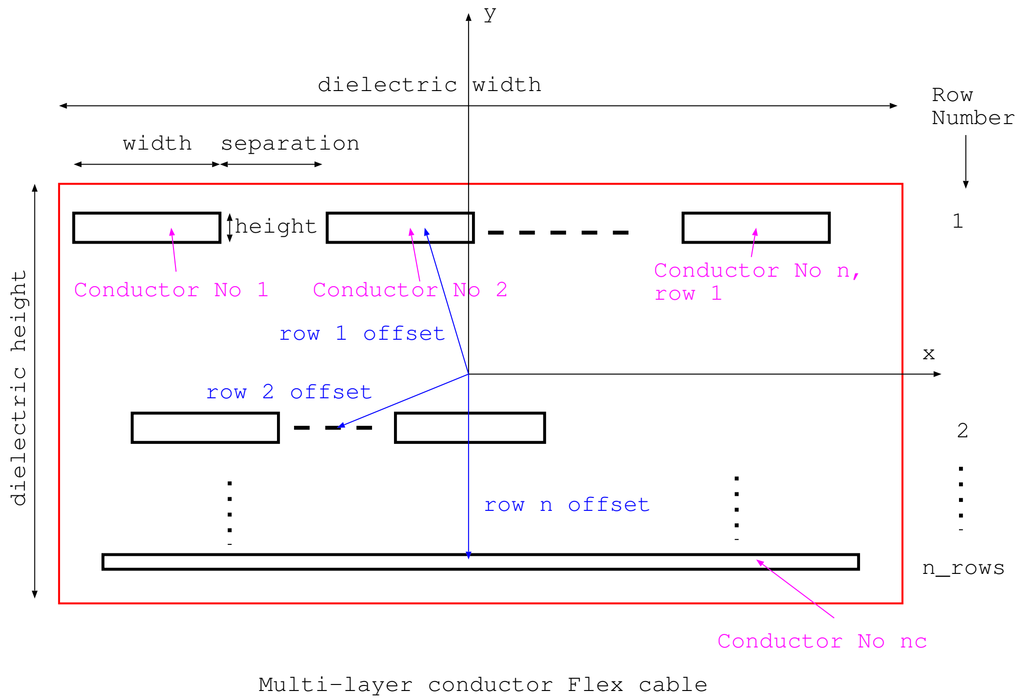


Figure 2.8 flex cable

example value	unit	Comment
8	integer	Number of conductors - can be any number of conductors in a flex cable model
6	integer	Number of parameters = (number of rows of conductors)*6+4
6.0e-3	metre	parameter 1: dielectric width (x)
1.2e-3	metre	parameter 2: dielectric height (y)
2	integer	parameter 3: number of rows of conductors
1.0e-3	metre	parameter 4: row 1 centre offset x
0.4e-3	metre	parameter 5: row 1 centre offset y
0.6e-3	metre	parameter 6: row 1 conductor width (x dimension)
1.0e-4	metre	parameter 7: row 1 conductor height (y dimension)
0.2e-3	metre	parameter 8: row 1 conductor separation
3	metre	parameter 9: row 1 number of conductors
1.0e-3	metre	parameter 10: row 2 centre offset x
-0.4e-3	metre	parameter 11: row 2 centre offset y
0.6e-3	metre	parameter 12: row 2 conductor width (x dimension)
1.0e-4	metre	parameter 13: row 2 conductor height (y dimension)
0.2e-3	metre	parameter 14: row 2 conductor separation
3	metre	parameter 15: row 2 number of conductors
5E7	Siemens/metre	parameter 16: conductivity
1	integer	number of frequency dependent parameters
$\epsilon\left(j\omega\right)$	rational function coefficients	Frequency dependent dielectric relative permittivity model

Table 2.8 Flex cable parameters

Example

```
#MOD_cable_lib_dir
.
flex_cable
3 # number of conductors
10 # number of parameters
5.0e-3          # parameter 1: dielectric width (x dimension)
1.25e-3   # parameter 2: dielectric height (y dimension)
1               # parameter 3: number of rows of conductors
0.0e-3   # parameter 4: row 1 centre offset x
0.0e-3   # parameter 5: row 1 centre offset y
1.0e-3          # parameter 6: row 1 conductor width (x dimension)
0.25e-3         # parameter 7: row 1 conductor height (y dimension)
0.5e-3          # parameter 8: row 1 conductor separation
3               # parameter 9: row 1 number of conductors
0.0           # parameter 10: conductivity
1 # number of frequency dependent parameters
# dielectric relative permittivity model follows
   1E9        # w normalisation constant
           0   # a order, a coefficients follow below:
  2.2
           0   # b order, b coefficients follow below:
  1.0
```

When creating cable bundle models including flex cables the Laplace solver must be used (include the line $use\_laplace$ following the bundle specification data in the $.bundle\_specfile$), and the mesh control parameter $Max\_mesh\_edge\_length$ may be used to give an additional degree of control over the Finite Element mesh. The number of mesh elements in the y direction on conductors (height) is set by the $Laplace\_surface\_mesh\_constant$ parameter (although the minimum number is two).

D connector

Figure 2.9 shows the cross section of the D connector model. A description of the model parameters is described in table 2.9 followed by an example. The D connector model differs from most other cable models in that the number of conductors is a parameter of the model. The conductors are numbered from left to right on the top row, then left to right on the bottom row and finally the D shaped shell conductor. The minimum number of conductors is 5.

The Laplace solution must always be used to calculate the per-unit-length parameters of the D-connector as there is no appropriate analytic solution available.

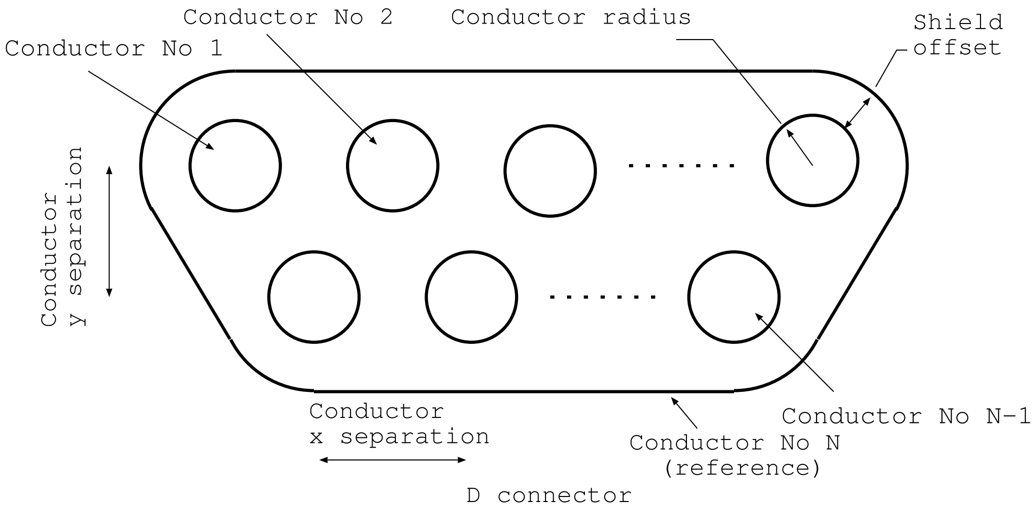


Figure 2.9 D connector

example value	unit	Comment
8	integer	Number of conductors - can be any number of conductors greater than 5
4	integer	Number of parameters, always 4 for D connectors
0.5e-3	metre	parameter 1: conductor radius
1.5e-3	metre	parameter 2: conductor pitch (separation in x)
1.5e-3	metre	parameter 3: conductor separation in y
1.0e-3	metre	parameter 4: offset from conductors to shell
0	integer	number of frequency dependent parameters
0	integer	number of transfer impedance models
$\epsilon(j\omega)$	rational function coefficients	Frequency dependent dielectric relative permittivity model

Table 2.9 D connector parameters

Example

```
#MOD_cable_lib_dir
.
Dconnector
10 # number of conductors
4 # number of parameters
0.5e-3   # parameter 1: conductor radius
1.5e-3   # parameter 2: conductor pitch (separation in x)
1.5e-3   # parameter 3: conductor separation in y
1.0e-3   # parameter 4: offset from conductors to shell
0 # number of frequency dependent parameters
0 # number of transfer impedance models
use_Laplace
```

Chapter 3

Creating a Cable Bundle Model

3.1 Introduction

This chapter describes the creation of a cable bundle model from previously defined cables. A cable bundle model is only concerned with the development of a multi-conductor propagation model for the bundle cross section and the calculation of the associated model parameters.

3.1.1 Cable bundle specification

A cable bundle is described as a combination of previously defined cables along with their position and orientation within the cross section of the bundle (x-y plane). A ground plane may also be included in the bundle specification. The ground plane is assumed to lie in the x-z plane i.e. along the x-axis in the cross section of the bundle.

The overshield cable type may enclose other cables within the bundle.

The cable position and orientation are defined using two or three real numbers. The first two are the x and y coordinates of the cable centre. The third number is a rotation angle for the cable, the angle is defined in an anticlockwise direction from the x axis. This is illustrated in figure 3.1 in which a coaxial cable is placed with its centre at point (x_1, y_1) and a flex cable is placed with its centre at point (x_2, y_2) and is rotated by an angle θ_2. A ground plane is also included in the bundle specification.

3.1.2 Conductor numbering within the bundle

The numbering of conductors within the bundle is very important for the correct use of the Spice cable bundle models as the conductor numbering within the bundle defines the connection nodes of the Spice subcircuit.

The numbering of conductors within a cable are shown in the figures for each cable type in section 2.

The conductors in the bundle are numbered in the order of their specification i.e. conductor number 1 is the first conductor of cable 1, conductor number 2 is the second

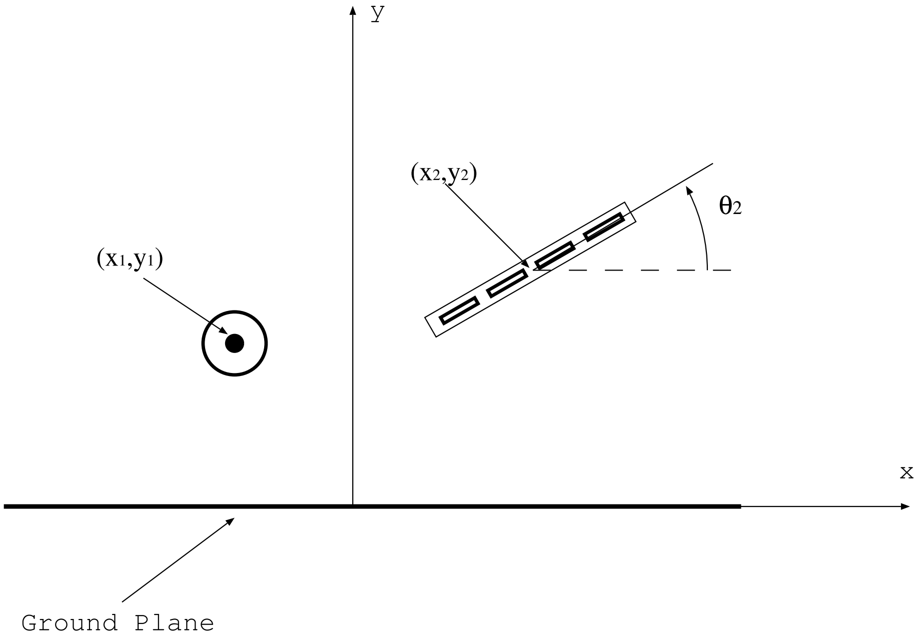


Figure 3.1 Specification of a cable bundle cross section

conductor of cable 1 etc until all the conductors of cable 1 are counted, the the next conductor is the first conductor of cable 2 and so on.

3.1.3 Cable bundle reference conductor

The reference conductor for the bundle is always the last conductor in the cable bundle specification. Note that for shielded cables the last conductor will always be the cable shield. If a ground plane is included in the bundle then this is always the last conductor to be specified and hence it automatically becomes the reference conductor for the bundle. The order of the cable specifications in the bundle will have no impact on the results from the model provided that the recommendations regarding the use of 'node zero' in section 9.3 are adhered to.

3.2 Cable Bundle Specification File Formats

This section describes the cable bundle specification file formats used as the input to the cable bundle model building process. Cable bundle specification files have the extension **name.bundle_spec**. The first lines of the **.bundle_spec** file specifies the path to the cable models used in the bundle specification (i.e. the path to the cable models within MOD), then the path to the cable bundle model to be produced from the specification (i.e. the path to the cable bundle models within MOD)

Following this is a list of cables which constitute the bundle and their configuration in the bundle cross section followed by an indication of the presence or absence of a ground plane.

If there are frequency dependent dielectrics present in the external domain of the bundle and it is required to take this frequency dependence into account in the bundle model then some 'filter fitting' information must be included so as to specify the order of the frequency dependent model to be used. This is the same information as that required to specify frequency depdendent models for shielded domains described in section 2.2.2. As for shielded domains, the filter fitting process provides a best fit model of specified order over a specified frequency range. As a default the model order is 0 i.e. no frequency dependence is included in the admittance matrix (the permittivity used is the high frequency value of the dielectric constant). The model order can be specified in two ways:

1. The order is specified as a positive integer and this is the order used
2. A negative integer is specified. In this case the order is chosen using an automatic algorithm which attempts to choose the best order from 0 up to |specified order|

following the model order the frequency range is specified. The frequency scale is set to be either linear ('lin') or logarithmic ('log'), following this the minimum frequency, maximum frequency and the number of frequencies for the filter fitting process are specified. If the Laplace solution is used then the number of frequencies should not be too large as this will lead to excessive runtimes for the cable model building process.

An example of a filter fitting specification for the admittance matrix element fit is shown below in which an order 2 model is specified based on a fit to logarithmic frequency data over a range of 1kHz to 1GHz with 12 sample points:

```
2          # order for admittance matrix element fit model
log          # frequency scale (log or lin)
1e3 1e9 12   # fmin fmax number_of_frequencies
```

In addition to the data required to specify a cable bundle, additional flags may be specified to influence the operation of the software. These flags are as follows:

1. 'verbose' output detailed summary of the software operation and calculation results.
2. 'use_Laplace' use the numerical Laplace solver to calculate inductance and capacitance matrices for the external domain and any overshielded domains. By default, approximate analytic formulae are used.
3. 'plot_mesh' output a vtk file which shows the mesh used in Finite Element Laplace calculations.

4. ‘direct_solver’ Solve the Finite Element matrix equation using a direct solver i.e. by calculating the matrix inverse.

5. ‘iterative_solver’ Solve the Finite Element matrix equation using an iterative solver. This is much more efficient than the direct solver and is the default option

6. ‘inf_gnd’ Include the ground plane as a boundary condition on the y=0 boundary in the Finite Element Laplace solution for the bundle cross section. the outer boundary in this case is a semi-circle.

7. ‘finite_gnd’ Include the ground plane as a finite conductor in the Finite Element Laplace solution for the bundle cross section. In this case the ground plane is a finite rectangular conductor within a circular boundary.

If the Laplace solver is used then the mesh generation is be controlled by the parameters

1. ‘Laplace_boundary_constant’ This parameter determines the distance to the outer boundary in open boundary domains. The distance to the outer boundary is calculated by first determining the largest dimension of the conductor system (including the origin if a ground plane is included), bundle_size. The outer boundary is defined as a circle of radius $bundle\_size * Laplace\_boundary\_constant$. The default value is 3.

2. ‘Laplace_surface_mesh_constant’ This parameter determines the number of finite element edges on a conductor surface. The edge length of elements on a cylindrical conductor of radius r is $\frac{r}{Laplace\_surface\_mesh\_constant}$. The default value is 3.

3. ‘Max_mesh_edge_length’ This (real) parameter determines the maximum finite element mesh edge length for the rectangular conductors of flex cables. It may be used to force an increase in mesh density in flex cables.

4. ‘gp_edge_length’ This (real) parameter determines the finite element mesh edge length at the centre of the ground plane. It may be used to increase the mesh density in the ground plane below the cable bundle to improve accuracy of the per-unit-length parameter calculation.

The default parameters are a compromise between accuracy and computation time for the Laplace solution. The default values may be overridden by the user by appending the following to the end of the **.bundle_spec** file:

```
Laplace_boundary_constant
4
Laplace_surface_mesh_constant
5
max_mesh_edge_length
0.02e-3
gp_edge_length
0.1e-3
```

Note that the Laplace solver must be used in cable bundles containing flex cables or D connectors.

If the iterative solver is used to solve the Finite Element matrix equations then the tolerance for the iterative solution may be set using the parameter cg_tol. The default value is 10^{-12}. The parameter may be set (in this example to 1E-14) by including the following lines int the **.bundle_spec** file:

```
cg_tol
1E-14
```

A ground plane may be included in the bundle as shown in figure 3.2

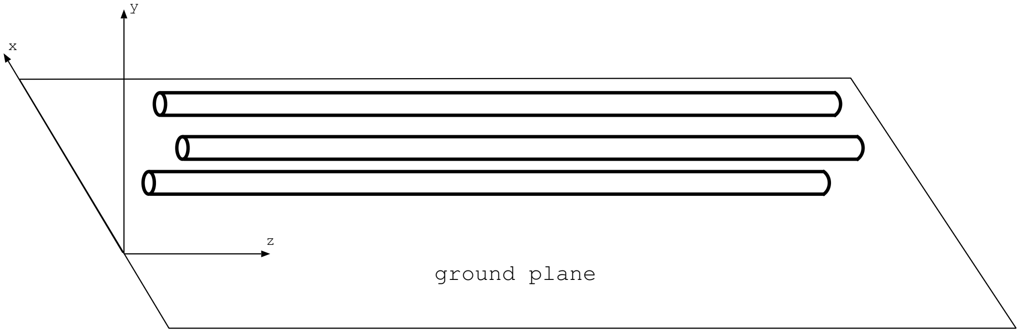


Figure 3.2 Specification of the ground plane position in the bundle cross section

The **.bundle_spec** file format is shown below, along with an example.

Line number	Typical value	Unit	Description
1	#MOD_cable_lib_dir	-	Comment line
2	.	-	Directory to read the cable model file from
3	#MOD_bundle_lib_dir	-	Comment line
4	.	-	Directory to write the cable bundle model file to
5	2	integer	Number of cables in the cable bundle
For each cable:			
-	cable name	-	Name of cable in the cable model directory
-	0.02 0.045 0.0	metres metres degrees	x and y coordinates of the centre of the cable in the bundle cross section and rotation angle of cable
-	ground_plane	-	ground_plane or no_ground_plane as required for the bundle

Example

Bundle model name: two_wires_over_ground

```
#MOD_cable_lib_dir
LIBRARY_OF_CABLE_MODELS
#MOD_bundle_lib_dir
LIBRARY_OF_BUNDLE_MODELS
2    # Number of cables in bundle, cable list follows
single_wire_with_FD_dielectric
6.35e-4     -0.001  0.0
single_wire
6.35e-4     0.001  0.0
ground_plane
2          # order for admittance matrix element fit (here an order 2 model)
log        # frequency scale for admittance matrix element fit (log or lin)
1e3 1e9 12 # fmin fmax number_of_frequencies for admittance matrix element fit
use_Laplace
plot_mesh
Laplace_boundary_constant
4
Laplace_surface_mesh_constant
5
```

Chapter 4

Creating a Spice Cable Bundle Model

This chapter describes the creation of a Spice cable bundle model from a previously defined cable bundle and additional specifications which are required to define the Spice model. This section also describes how a validation test case is defined which enables the performance of the model to be assessed in either an a.c. or transient analysis.

The format of the files relating to the Spice cable bundle model building process are described in section 4.4.

4.1 Incident field excitation

The transmission line model may incorporate an incident field excitation which takes the form of a plane wave. The plane wave incident field may be a continuous wave for a.c. analysis or a pulse for transient analysis. When a plane wave excitation is required the Spice sub-circuit model has an additional two nodes defined, the voltage between these nodes specifies the Electric field of the plane wave.

For the purposes of incident field excitaton analysis the axis of the cable bundle is assumed to be in the z direction. The direction of the incident field plane wave is specified by its wave vector in polar coordinates (figure 4.1)

$$\begin{aligned} k_x &= \sin(k_\theta)\cos(k_\phi) \\ k_y &= \sin(k_\theta)\sin(k_\phi) \\ k_z &= \cos(k_\phi) \end{aligned} \tag{4.1}$$

where k_θ is the angle of the wave vector from the z axis and k_ϕ is the angle from the x axis as shown in figure 4.2.

Given this k vector, unit vectors in the θ and ϕ directions can be identified. The Electric field polarisation direction is specified in terms of these vectors (E_θ and E_ϕ) as shown in figure 4.2. It is important to note that the angles E_θ and E_ϕ specify the polarisation direction only, the amplitude of the Electric field is specified separately i.e.

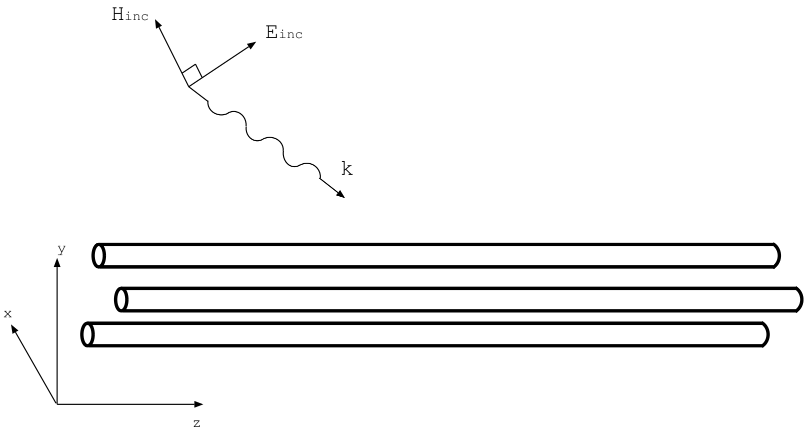


Figure 4.1 Plane wave incident field excitation of a cable bundle

the polarisation vector is normalised to unit length and only the relative amplitudes of E_θ and E_ϕ is significant in determining the polarisation direction so setting $E_\theta = 1$ and $E_\phi = 1$ will give the polarisation vector as specifying $E_\theta = \frac{1}{\sqrt{2}}$ and $E_\phi = \frac{1}{\sqrt{2}}$.

A wave in the +z direction with the Electric field polarised in the +x direction is therefore specified by $k_\theta = 0°$ $k_\phi = 0°$ $E_\theta = 1$ $E_\phi = 0$

A wave in the +z direction with the Electric field polarised in the +y direction is therefore specified by $k_\theta = 0°$ $k_\phi = 90°$ $E_\theta = 1$ $E_\phi = 0$

A wave in the +x direction with the Electric field polarised in the +z direction is therefore specified by $k_\theta = 90°$ $k_\phi = 0°$ $E_\theta = -1$ $E_\phi = 0$

A wave in the -y direction with the Electric field polarised in the +z direction is therefore specified by $k_\theta = 90°$ $k_\phi = -90°$ $E_\theta = -1$ $E_\phi = 0$

A wave in the -y direction with the Electric field polarised in the +x direction is therefore specified by $k_\theta = 90°$ $k_\phi = -90°$ $E_\theta = 0$ $E_\phi = 1$

The derivation of the Spice cable bundle model for incident field excitation requires that $0° \leq k_\theta \leq 90°$. If k_θ is out of this range then an error will occur. It should be noted that models for the situation $k_\theta > 90°$ can be simulated by swapping the ends to which the termination circuits are connnected.

4.2 Transfer impedance model

The Spice cable bundle model is based on a domain decomposition method where a domain is formed by the set of conductors within a shield separating them from a possible external domain (if it exists).

The domains are modelled independently i.e. signals propagate within domains and do not couple between domains through cable shields unless specifically requested. The Spice cable bundle model uses a weak form of transfer impedance coupling thus a source and a victim domain must be specified for the inclusion of a transfer impedance model.

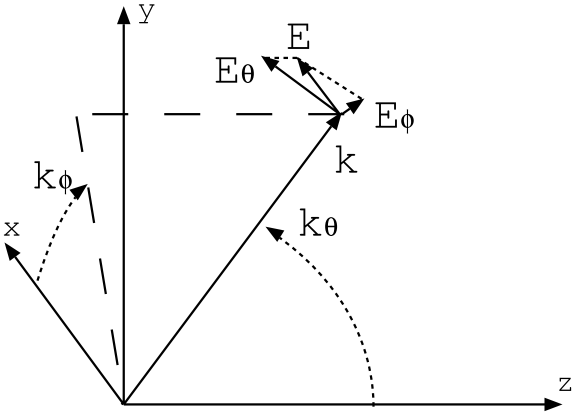


Figure 4.2 Specification of the incident field polarisation

The transfer impedance of cable shields and overshields will have been specified in the individual cable models as a conductor impedance model for a shield (section 2.2.1). This will automatically be included as a loss in the propagation algorithm for signals propagating on the inside and outside of the shield. The model extracts the d.c. resistance of conductors and includes this directly on each conductor therefore coupling between domains does exist in the model due to the d.c. transfer impedance. However as a default no coupling between the internal and external domains is included apart from that which results from the d.c. component of the transfer impedance.

In order to include a transfer impedance model the conductor number of the shield of interest must be specified along with the direction of the coupling. The coupling direction is specified in the **name.spice_model_spec** file as either +1 or -1 where +1 indicates coupling direction from inside the shield to outside and -1 indicates coupling from the outside to the inside.

An example of the specification of transfer impedance coupling within a Spice cable bundle model is seen in section4.4

4.3 Spice cable bundle subcircuit node numbering

The subcircuit node numbering is as follows:

1. End 1 (- z end) conductors in order of the conductors in the bundle specification
2. End 2 (+ z end) conductors in order of the conductors in the bundle specification
3. If an incident field excitation is required, there are two additional nodes to specify the incident (Electric) field excitation function. The incident electric field is equal to the voltage between these two nodes.

4.4 Spice Cable Bundle Specification File Formats

This section describes the Spice cable bundle specification file formats used as the input to the Spice cable bundle model building process. Spice cable bundle specification files have the extension **.spice_model_spec**.

The first lines of the **.spice_model_spec** file specifies the path to the cable models used in the cable bundle specification (i.e. the path to the cable models within MOD), then the path to the cable bundle models used in the Spice model (i.e. the path to the cable bundle models within MOD) then the path and to the Spice model to be produced (i.e. the path to the Spice cable bundle models within MOD) and finally the path to the Spice model circuit symbols.

The input file to the Spice cable bundle building processs then includes the bundle name, bundle length, incident field specification (if required) and the specification of the validation test configuration (if required).

In addition information regarding the transfer impedance models to be included and also information to control the transfer function fitting process can be specified.

The validation test configuration is shown in figure 4.3

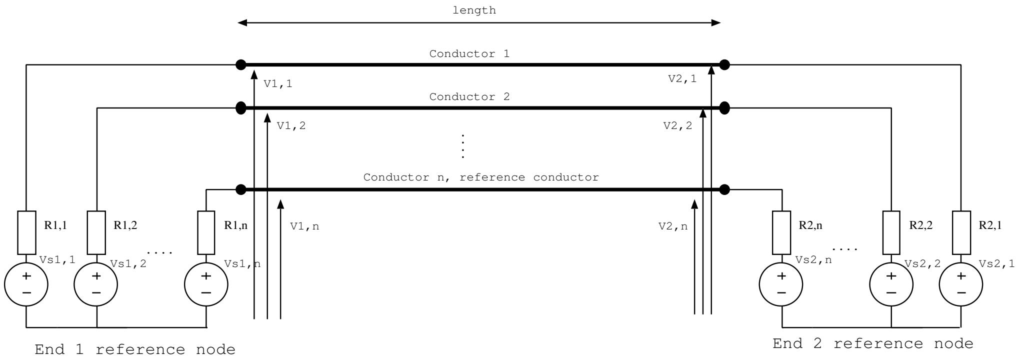


Figure 4.3 Validation test case configuration

In addition to the data required to specify a Spice cable bundle model, additional flags may be specified to influence the operation of the software. These flags are as follows:

1. ‘verbose’ output detailed summary of the software operation and calculation results.

2. ‘use_xie’ Include direct incident field to shielded domain coupling terms (Xie’s model). This is the default.

3. ‘no_xie’ Do not include direct incident field to shielded domain coupling terms

4. 'use_high_freq_zt_model' Use a distributed model for the transfer resistance - this may give improved transfer impedance coupling results at high frequency for cases when the shield termination impedance is very small at the expense of low frequency accuracy. This is the default.

5. 'no_high_freq_zt_model' Use a lumped model for the transfer resistance - this may give poor transfer impedance coupling results at high frequency for cases when the shield termination impedance is very small however it will give good results at low frequency. This is the default state as it is expected that once realistic bonding impedances are taken into account, the termination impedance will not be small enough to cause a problem.

6. 'use_s_xfer' Use the s-domain transfer function to implement the frequency dependent transfer function used in the propagation correction and transfer impedance implementation.

7. 'no_s_xfer' Use a lumped component model to implement the frequency dependent transfer function used in the propagation correction and transfer impedance implementation. This is now the default state of this flag following the development of the equivalent circuit model for rational s-domain transfer functions of arbitrary order. This is the default.

8. 'use_LTRA' Implement delay lines in the Spice models as LTRA transmission lines rather than simple delay lines (T elements). This is found to improve the robustness of the Spice transmission line models, esepcially for transient analysis in Ngspice. This is the default state for this flag.

9. 'no_LTRA' Implement delay lines in the Spice models as T elements rather than LTRA transmission lines.

There are also a number of constants used in the development of the models which may be changed from their default values from within the **.spice_model_spec** file. The format is a line with the constant name and the following line has the new value. The constants which may be specified by the user are:

1. 'min_delay' . Minimum delay allowed for transmission lines in the sub-circuit models. Transmission lines with delays less than this value are dealt with in a different manner. The default value is 10^{-12} s.

2. 'Rsmall' Minimum resistance value in the sub-circuit model. Resistances less than this value are replaced by this value. The default value is $10^{-8}\Omega$.

The file format with no transfer impdance models or transfer function fitting information is as follows:

Line number	Typical value	Unit	Description
1	#MOD_cable_lib_dir	-	Comment line
2	.	-	Directory to read the cable model file from
3	#MOD_bundle_lib_dir	-	Comment line
4	.	-	Directory to read the cable bundle model file from
5	#MOD_spice_bundle_lib_dir	-	Comment line
6	.	-	Directory to write the Spice cable bundle model file to
7	# two wires over ground plane, crosstalk model	-	Comment line
8	two_wires_over_ground	-	Cable bundle name Note there should be nothing else on this line
9	# bundle length	-	Comment line
10	2.0	metres	Cable bundle length
11	# incident field specification	-	Comment line
12	0.0	V/m	Amplitude
13	90.0 0.0	degrees	Wave vector angle kθ kϕ
14	1.0 0.0	degrees	Polarisation Eθ Eϕ
15	# End 1 termination model	-	Number of sources and resistances = number of conductors
16	1.0	V	End 1, conductor 1 voltage source amplitude
-	0.0	V	End 1, conductor 2 voltage source amplitude
-	0.0	V	End 1, reference conductor voltage source amplitude
-	50.0	Ω	End 1, conductor 1 resistance
-	25.0	Ω	End 1, conductor 2 resistance
-	0.0	Ω	End 1, reference conductor resistance
-	# End 2 termination model	-	Number of sources and resistances = number of conductors
-	1.0	V	End 2, conductor 1 voltage source amplitude
-	0.0	V	End 2, conductor 2 voltage source amplitude
-	0.0	V	End 2, reference conductor voltage source amplitude
-	50.0	Ω	End 2, conductor 1 resistance
-	25.0	Ω	End 2, conductor 2 resistance
-	0.0	Ω	End 2, reference conductor resistance

Line number	Typical value	Unit	Description
-	# Type of analysis	-	Comment line
-	AC	-	AC or TRANS
	For AC analysis		
-	lin	-	logarithmic (log) or linear (lin) frequency scale
-	1e3 1e8 1000	Hz Hz integer	min frequency, max frequency number of frequencies
	For TRANSIENT analysis		
-	0.01E-9 100E-9	seconds seconds	# timestep runtime
-	1e-9 50e-9	seconds seconds	# pulse risetime pulse width
-	# Output_conductor1 output_conductor2 and end number	-	Conductor_2 can be left out in which case the voltage is relative to the reference conductor
-	1 1	integer integer	Output conductor number and end number
-	lin	-	For AC only: Output voltage scaling, linear or dB

AC example

An example file for the AC analysis of two wires over a ground plane is shown below:

```
#MOD_cable_lib_dir
LIBRARY_OF_CABLE_MODELS
#MOD_bundle_lib_dir
LIBRARY_OF_BUNDLE_MODELS
#MOD_spice_bundle_lib_dir
./
#spice_symbol_dir
SYMBOL_DIR
# Specification for Spice model of two wires over ground
two_wires_over_ground
# cable bundle length (m)
2.0
#Incident field specification
0.0 amplitude (V/m)
90.0 0.0 ktheta kphi (degrees)
-1.0  0.0 Etheta Ephi
# End 1 termination model
1.0      End 1 voltage source list
0.0
0.0
100.0    End 1 impedance list
25.0
0.0
# End 2 termination model
0.0      End 2 voltage source list
0.0
0.0
75.0    End 2 impedance list
50.0
0.0
# Type of analysis
AC
log          # frequency scale (log or lin)
1e3 1e8 1000 # fmin fmax number_of_frequencies
# Output conductor number and end number
1    1
lin   # output type (lin or dB)
```

Transient example

An example file for the trasnsient analysis of two wires over a ground plane is shown below:

```
#MOD_cable_lib_dir
.
#MOD_bundle_lib_dir
.
MOD_spice_bundle_lib_dir
.
#spice_symbol_dir
SYMBOL_DIR
2   # number of conductors, n.
2_wire
# cable bundle length (m)
2.0
#Incident field specification
0.0 amplitude (V/m)
90.0 0.0 ktheta kphi (degrees)
-1.0  0.0 Etheta Ephi
# End 1 termination model
1.0      End 1 voltage source list
0.0
50.0   End 1 impedance list
0.0
# End 2 termination model
0.0        End 2 voltage source list
0.0
50.0    End 2 impedance list
0.0
# Type of analysis
TRANS
0.01E-9 100E-9   # timestep runtime
1e-9 50e-9 # pulse_risetime pulse_width
# Output conductor number and end number
1  1
```

In the case of frequency dependent models and transfer impedance coupling models then the above file format can be adapted to include additional information required to drive these models.

A weak form of the transfer impedance is implemented in this work hence the direction of coupling must be specified. In order to include a transfer impedance model the conductor number for the shield whose transfer impedance is required. In addition to this the direction of the transfer impedance coupling must be specified. The direction is specified as an integer where +1 indicates coupling from inside the shield to outside and -1 indicates coupling from the outside to the inside.

The frequency dependent propagation correction takes the form of a s-domain transfer function in the Spice model. These propagation correction transfer functions are derived using a rational function fitting process. This process provides a best fit model of specified order over a specified frequency range. As a default the model order is 0 i.e. no frequency dependent propagation correction. The model order can be specified in two ways:

1. The order is specified as a positive integer and this is the order used

2. A negative integer is specified. In this case the order is chosen using an automatic algorithm which attempts to choose the best order from 0 up to $|\text{specified order}|$

The frequency range for the model fit may also be specified as can the use of a log or linear frequency scale. If the frequency range is not specified then it is derived from the definition of the validation test case.

The format is descibed below followed by an example.

Line number	Typical value	Unit	Description
1	#MOD_cable_lib_dir	-	Comment line
2	.	-	Directory to read the cable model file to
3	#MOD_bundle_lib_dir	-	Comment line
4	.	-	Directory to read the cable bundle model file from
5	#MOD_spice_bundle_lib_dir	-	Comment line
6	.	-	Directory to write the Spice cable bundle model file to
7	# two wires over ground plane, crosstalk model	-	Comment line
8	two_wires_over_ground	-	Cable bundle name Note there should be nothing else on this line
9	# bundle length	-	Comment line
10	2.0	metres	Cable bundle length
11	# incident field specification	-	Comment line
12	0.0	V/m	Amplitude
13	90.0 0.0	degrees	Wave vector angle kθ kϕ
14	1.0 0.0	relative amplitudes	Polarisation Eθ Eϕ
15	#Transfer impedance terms	-	this line must inlcude the words ‘transfer impedance’ to indicate that transfer impedance information follows
16	1	integer	# number of transfer impedances to include in the model
-	3 +1	integer integer	# shield conductor number and coupling direction for transfer impedance model 1 + is inside to out
-	# End 1 termination model	-	Number of sources and resistances = number of conductors
-	1.0	V	End 1, conductor 1 voltage source amplitude
-	0.0	V	End 1, conductor 2 voltage source amplitude
-	0.0	V	End 1, reference conductor voltage source amplitude
-	50.0	Ω	End 1, conductor 1 resistance
-	25.0	Ω	End 1, conductor 2 resistance
-	0.0	Ω	End 1, reference conductor resistance

Line number	Typical value	Unit	Description
-	# End 2 termination model	-	Number of sources and resistances = number of conductors
-	1.0	V	End 2, conductor 1 voltage source amplitude
-	0.0	V	End 2, conductor 2 voltage source amplitude
-	0.0	V	End 2, reference conductor voltage source amplitude
-	50.0	Ω	End 2, conductor 1 resistance
-	25.0	Ω	End 2, conductor 2 resistance
-	0.0	Ω	End 2, reference conductor resistance
-	# Type of analysis	-	Comment line
-	AC	-	AC or TRANS
	For AC analysis:		
-	lin	-	logarithmic (log) or linear (lin) frequency scale
-	1e3 1e8 1000	Hz Hz integer	min frequency, max frequency number of frequencies
	For TRANSIENT analysis		
-	0.01E-9 100E-9	seconds seconds	# timestep runtime
-	1e-9 50e-9	seconds seconds	# pulse risetime pulse width
-	# Output conductor number and end number	-	Comment line
-	1 1	integer integer	Output conductor number and end number
-	lin	-	For AC only: Output voltage scaling, linear or dB
-	-10	integer	# order for transfer function fit model
-	log	-	logarithmic (log) or linear (lin) frequency scale for transfer function fitting
-	real real integer	1e5 1e9 200	# fmin fmax number_of_frequencies for transfer function fitting

AC example with a transfer impedance model

This example demonstrates a model of coupling from a single wire to a coaxial cable. The bundle is defined such that conductor 1 is the single wire, conductor 2 is the inner coax conductor and conductor 3 is the coaxial cable shield.

```
#MOD_cable_lib_dir
./
#MOD_bundle_lib_dir
./
#MOD_spice_bundle_lib_dir
./
#spice_symbol_dir
./
# Specification for Spice model of two wire transmission line, no loss
zt_test
# cable bundle length (m)
1.0
#Incident field specification
0.0 amplitude (V/m)
90.0 0.0 ktheta kphi (degrees)
-1.0  0.0 Etheta Ephi
#Transfer impedance terms
1    # number of transfer impedances to include in the model
3 +1 # conductor number and coupling direction for transfer impedance model
# End 1 termination model
 0.02      End 1 voltage source list
 0.5
 0.0
300.0   End 1 impedance list
20.0
0.0
# End 2 termination model
0.03        End 2 voltage source list
1.0
0.0
1000.0   End 2 impedance list
150.0
0.0
# Type of analysis
AC
log          # frequency scale (log or lin)
1e5 1e9 1000 # fmin fmax number_of_frequencies
# Output conductor number and end number
1    2
lin   # output type (lin or dB)
-10     # order for transfer function fit model
```

```
log          # frequency scale for transfer function fit (log or lin)
1e5 1e9 200 # fmin fmax number_of_frequencies for transfer function fit
```

Chapter 5

Library of cable models, MOD

MOD comprises the library of cable models; the library contains the following:

1. Models of individual cables which are created by the cable model building process or delivered as elements of the library. These models may be used as input to the cable bundle model building process.

2. Models of cable bundles which are created by the cable bundle model building process and may be used as input to the Spice cable bundle model building process.

3. Spice models of specific cable bundle modelling configurations including length, incident field excitation, transfer impedance coupling and target Spice version. These models will have an accompanying symbol corresponding to the target Spice version and are created by the Spice cable bundle model building process. These models may then be included in appropriate versions of Spice.

The library of cable models is a hierarchical structure, this structure reflects the manner in which the elements which go to create a Spice cable bundle model are combined.

The library contains a directory of individual cable models (**CABLE**) e.g. single conductors, twisted pairs, coaxial cables, shielded twisted pairs etc. These cable models may be combined in any configuration, along with ground plane and overshield specifications to form a cable bundle model. A cable bundle model contains all the information required to characterise a particular bundle configuration. Cable bundle models are kept in the directory **BUNDLE**. Cable bundle models are identified by a name provided by the user. A cable bundle may be applied in a range of modelling scenarios, each requiring a separate Spice model due to differences in configuration (cable bundle length incident field excitation, transfer impedance coupling models included, and their coupling direction etc.) Each of the cable bundle models within the **BUNDLE** directory can therefore be used as the basis of a number of Spice cable bundle models. These Spice cable bundle models may be kept in the **SPICE** directory within MOD. This directory also contains a sub-directory **SYMBOL** containing the circuit symbols for the Spice cable bundle models. These circuit symbols may be exported to the appropriate place for use with gschem or LTspice as described in chapter 9.

The software can produce Spice sub-circuit models of transmission lines for three versions of Spice:

1. Ngspice
2. LTspice
3. PSpice

The file formats for the three Spice versions differ so MOD can contain up to three different versions of the same Spice subcircuit model.

Chapter 6

Running the test cases

The TEST_CASES directory contains a number of test cases which are designed to test the functionality of the Spice cable model building project.

The script **generate_spice_cable_bundle_model** automatically generates Spice cable bundle models, validation test circuits and analytic results for multi-conductor cable test cases. The Spice model results are plotted against the analytic solution for both frequency domain and transient analyses. The process also calculates a figure of merit for the model based on the difference between the analytic and Spice model data using the code **compare_results**.

In order to run the test cases you will need the following to be installed: Ngspice gnuplot

The script may need to be changed for your system. It is set up by default to work with a library of cable models in the TEST_CASES directory. This may be changed to suit the user by for example changing the path for the library of cable models (MOD) by changing the variable in the **generate_spice_cable_bundle_model** script:

LIBRARY_OF_MODELS_TOP_LEVEL="/home/user/LIBRARY_OF_CABLE_MODELS"

The user may also want to specify a directory for the circuit symbols produced by the system. The default action is to keep these in the directory where the cable models are developed. This may be changed by setting the variable SYMBOL_DIR in the script **generate_spice_cable_bundle_model**. For example for use by gshem you may need to following:

SYMBOL_DIR="/usr/share/gEDA/sym/local"

(NOTE: you may need to change ownership of **SYMBOL_DIR** with something like the following command: **sudo chown chris:chris /usr/share/gEDA/sym/local**)

Test cases are found in sub-directories in the TEST_CASES directory. In each test case directory the following files should exist:

1. A number of files (or zero if they already exist in the library) for cable specification information ***.cable_spec**

2. Zero or a maximum of one file for cable bundle specification ***.bundle_spec**

3. One (and only one) file for the Spice cable bundle specification '***.spice_model_spec**

The automatic testing script is run using the following command:
generate_spice_cable_bundle_model action
action is the process to perform:

action=run NAME	Generate cable models, bundle model and Spice bundle model and validation Spice model as required for the test case NAME. Run Ngspice on the validation model and compare the Spice result against the analytic solution
action=plot NAME	Plot the results of the Spice validation model and analytic solution to screen
action=plot_wxt NAME	Plot the results of the Spice validation model and analytic solution to wxt terminal
action=plot_jpg NAME	Plot the results of the Spice validation model and analytic solution to .jpg file
action=plot_ref NAME	Plot the Spice validation model results against reference results to x11 terminal
action=plot_ref_wxt NAME	Plot the Spice validation model results against reference results to wxt terminal
action=plot_ref_jpg NAME	Plot the Spice validation model results against reference results to jpg file
action=plot_bundle NAME	Plot the bundle cross section to the screen
action=plot_bundle_wxt NAME	Plot the bundle cross section to wxt terminal
action=plot_bundle_jpg NAME	Plot the bundle cross section to jpg file
action=plot_bundle_png NAME	Plot the bundle cross section to png file
action=plot_bundle_svg NAME	Plot the bundle cross section to svg file
action=reference NAME	Update the reference results and the difference between the Spice model and validation model with the current set of results i.e. the results of the latest run.
action=check_error NAME	Quick check that the difference between the Spice model and validation model is the same as for the reference data set
action=clean NAME	Remove all files except those required to set up the problem and the reference results.
action=clean_all NAME	Remove all files including the reference results.

NAME can be one of the existing test cases or left blank to process all of the test cases

The test case directory should contain the following:

***.cable_spec** files as required to specify cables (though cables specified elsewhere could be used to form bundles)

a ***.bundle_spec** file if required to specify the cable bundle (though cable bundles specified elsewhere could be used in the specification of Spice cable models)

a ***.spice_model_spec** file to specify the Spice cable bundle model if a Spice model is to be generated for this test case.

The action ‘run’ does the following:

1. copy the input files which describe the required simulation to the **RUN_DIRECTORY** directory. The input files are as follows: ***.cable_spec** file(s) which describe individual cable types, a ***.bundle_spec** file which describes the cable bundle cross section and a ***.spice_model_spec** file which describes the simulation scenario and a validation test configuration.

2. run the code **cable_model_builder** as required to create the cable models. the output of this process is a cable model in a file (***.cable**) corresponding to each of the cable specification files (***.cable_spec**).

3. run the code **cable_bundle_model_builder** as required to create the cable bundle models. The output of this process is a cable bundle model in a file (***.bundle**).

4. run the code **spice_cable_bundle_model_builder** as required to create

   a. Spice subcircuit models of the bundle for each of the target Spice versions (***_Ngspice.lib**, ***_LTspice.lib**, ***_PSpice.lib**).

   b. the schematic symbol for the Spice subcircuit model (***.sym**)

   c. Spice validation models for each of the target Spice versions (***_Ngspice.cir**, ***_LTspice.cir**, ***_PSpice.cir**).

   d. the analytic solution of the validation configuration (**analytic_solution.dat**)

5. run **Ngspice** on the Spice validation model to produce the Spice solution output (**spice_solution.dat**)

6. run the code **compare_results** which provides a numerical measure of the difference between the results of the analytic solution and the Spice solution (**result_comparison.dat**).

   For frequency domain simulations the difference measure is calculated as

   $$err = \frac{\int_{fmin}^{fmax} |V_{analytic}(f) - V_{spice}(f)|\, df}{fmax - fmin} \tag{6.1}$$

   For time domain simulations the difference measure is calculated as

   $$err = \frac{\int_{tmin}^{tmax} |V_{analytic}(t) - V_{spice}(t)|\, dt}{tmax - tmin} \tag{6.2}$$

Chapter 7

Running the software without the GUI

There are three processes in the creation of a Spice cable bundle model. These are:

1. cable model building process
2. cable bundle model building process
3. Spice cable bundle model building process

In addition there are the supporting processes to generate frequency dependent dielectric models from tabulated frequency domain relative permittivity data and togenerate frequency dependent transfer impedance models from a description of the braid geometry.

In this chapter the inputs and outputs to these processes are outlined. The detail of the input file formats for the Spice cable bundle creation process may be found in chapters 2, 3 and 4.

7.1 Cable model building process

The inputs to the Cable model building process enable the characterisation of individual cables. For a particular cable type all the information required to model the cable within a bundle must be supplied. A cable will be characterised by its cross section geometry and material parameters. The extension for cable specification information files is ***.cable_spec**. The cable types available are:

1. Cylindrical conductor with dielectric
2. Coaxial cable with transfer impedance and shield surface impedance loss
3. Twinax cable with transfer impedance and shield surface impedance loss

4. Twisted pair
5. Shielded twisted pair with transfer impedance and shield surface impedance loss
6. Spacewire with transfer impedance and shield surface impedance loss
7. Overshield with transfer impedance and shield surface impedance loss
8. flex cable
9. D connector

The inputs to the cable model building process are as follows:

1. Geometric and (constant with frequency) material parameters for the cable as requred for each cable type
2. Any frequency dependent dielectric models required for the cable type
3. Any frequency dependent transfer impedance models required for the cable type
4. Flags to control the operation of the software.
5. Mesh parameters for Laplace solution (if required):

The detail of the ways in which a cable is specified are detailed in section 2 along with the format of the information in the ***.cable_spec** file.

To run the cable model building process use the command:

cable_model_builder

The user is prompted to enter the name of the cable specification data file (without **.cable_spec** extension) i.e. there must be an existing file **name.cable_spec** containing the cable specification. Alternatively the user can supply the name of the cable as a command line argument i.e.

cable_model_builder cable_name

This is the only action required by the user.

The output of the cable model building process is a fully specified cable model in a file (**name.cable**).

The cable models include internal propagation characterization (L, C matrices), shield characterization and loss model parameters also the domain decomposition matrices for shielded cables.

These outputs can be used to populate the library of cable models (MOD) with cable models by specifying an appropriate MOD directory in the ***.cable_spec** file.

7.2 Cable bundle model building process

The cable bundle model consists of the propagation models of the shielded domains which have already been characterised on a cable basis. In addition to this the propagation on the bundle requires the external domain to be modelled, plus any domains defined by the presence of over-shields. The extension for cable bundle specification information files is ***.bundle_spec**.

The configuration of the external domain is characterised by the geometry of the external conductors and dielectrics for each cable in the bundle, the geometric configuration of the cables within the bundle and its relation to the ground plane (if it exists). Similarly, domains within over-shields are characterised by the geometry of the external conductors and dielectrics for each cable within the shielded domain. Thus the cross section geometry of the cable bundle must be completely specified at this stage, including ground plane and/or overshields.

The inductance and capacitance matrices for the external and over-shielded domains may be determined from a numerical Laplace equation solution applied to the appropriate domain geometry or an approximate analytic solution according to flags set by the user. Loss models are based on models of skin effect appropriate for the conductor geometry. Details of the Laplace equation solution and loss models may be found in the Theory Manual [3].

The inputs to the cable bundle model building process are as follows:

1. Cable models output from the cable model building process which may be obtained from the library of cable models (MOD) or elsewhere.

2. Bundle cross section geometry i.e. the placement of the individual cables in relation to each other in the bundle cross section. the bundle geometry can include overshields.

3. Ground plane specification (if required). The ground plane is assumed to be situated along the x axis in the x-y cross section of the cable bundle.

4. Flags to control the operation of the software.

5. Mesh parameters for Laplace solution (if required):

The detail of the ways in which a cable bundle is specified are detailed in section 3 along with the format of the information in the ***.bundle_spec** file.

To run the cable bundle model building process use the command:

cable_bundle_model_builder

The user is prompted to enter the name of the cable specification data (without **.bundle_spec** extension). Alternatively the user can supply the name of the bundle as a command line argument i.e.

cable_bundle_model_builder bundle_name

This is the only action required by the user.

The output from the cable bundle model building process is a cable bundle model (***.cable_bundle**) file. This file incorporates a model of the cable bundle decomposed into domains. The model specifies the cables forming the bundle, the bundle cross section and decomposition of the cable bundle into domains i.e. which cables and conductors belong in which domain, which conductors (shields) separate which domains, transfer impedance models for shields. Within each domain the following is specified:

1. Inductance and capacitance matrix

2. Loss model
3. Domain decomposition matrices

The output can be used to populate the library of cable models (MOD) with cable bundle models by specifying an appropriate MOD directory in the ***.bundle_spec** file.

7.3 Spice cable bundle model building process

Once a cable bundle has been specified a Spice cable model can be created for the bundle. The Spice model will necessarily be dependent on the particular analysis required for a bundle for example different incident field excitations may be specified or transfer impedance coupling paths included in the model. Also note that the Spice cable models are not portable between different versions of Spice.

The Spice cable bundle model building process inputs are as follows:

1. Bundle model which is characterised by the inductance and capacitance matrices in each domain and the conductor loss models, frequency dependent transfer impedances of shields, spatial configuration of conductors in the external domain and domain decomposition matrices. The bundle model can come from the library of cable models, MOD.
2. Bundle length.
3. Incident field excitation (angle, polarization).
4. Source and victim conductors for transfer impedance coupling.
5. Frequency range for model.
6. Spice version required.
7. Validation test case configuration.
8. Flags to control the operation of the software.
9. Constants to be changed from their default values.

The Spice cable bundle model building process output consists of the following:

1. The Spice Cable Model and an associated schematic symbol
2. Spice cable model with a test/validation configuration.

3. Validation data.

4. The Spice cable model and schematic symbol can form an input to the library of cable models (MOD).

The ways in which a Spice cable bundle is specified are detailed in section 4 along with the format of the information in the ***.spice_model_spec** file.

To run the cable bundle model building process use the command:

spice_cable_bundle_model_builder

The user is prompted to enter the name of the Spice cable bundle model specification data (without **.spice_model_spec** extension). Alternatively the user can supply the name of the Spice cable bundle model as a command line argument i.e.

spice_cable_bundle_model_builder spice_cable_bundle_model_name

This is the only action required by the user.

7.3.1 Transient validation test cases

When setting up a transient validation test case it is extremely important that the runtime specified is sufficient for all the transients in the simulation to reduce to an insignificant level. If this is not the case then significant errors can arise in the analytic transient solution due to aliasing in the FFT implementation of the convolution process.

7.4 Generation of frequency dependent rational function models

In order to take full advantage of the frequency dependent models available it may be necessary to generate a rational function representation of a frequency dependent parameter. This may be the relative permittivity of a dielectric material or a transfer impedance. A command line driven process, **rational_function_fit**, is available which allows the calculation of rational function coefficients from tabulated complex frequency domain data. The process provides a best fit model of specified order over the specified frequency range. The model order can be specified in two ways:

1. The order is specified as a positive integer and this is the order used

2. A negative integer is specified. In this case the order is chosen using an automatic algorithm which attempts to choose the best order from 0 up to |specified order|

The frequency range for the model fit must be specified as must be the use of a log or linear frequency scale.

The complex frequency domain data should be in a file formatted with three columns for the frequency, real part and imaginary part of the function respectively.

To run the rational function fitting process use the command:

rational_function_fit

The user is prompted to enter the name of the complex frequency domain data file (**file_name**). The file is then read before the user is prompted to provide the numerator order and the denominator order for the rational function. The best fit model is calculated and the stability of the resulting model is determined (i.e. whether the impulse response of the transfer function converges).

The output of the process is a file **name.function_fit** which contains the rational function model fit to the input function. In addition a file **function_fit.fout** is produced which contains tfive columns; frequency, Real part of the input function data, Imaginary part of the input function data, Real part of the function fit data, Imaginary part of the function fit data. This data set allows the original data and the model fit data to be visualised.

7.5 Generation of frequency dependent transfer impedance models

The program **shield_conductor_and_transfer_impedance_model_builder** calculates the frequency dependent transfer impedance and surface impedance of a braided shield from the shield specifications using the theory described by Vance ¡citevance and Kley [7] and summarised in appendix A. The braid specification is given in a file **name.braid_spec**. The braid specification consists of the parameters given in table 7.1.

example value	unit	Comment
0.00294	m	braid diameter, D (m)
8	integer	Number of carriers, C
4	integer	Number of wires in a carrier, N
0.25e-3	m	diameter of a single wire, d (m)
5E7	S/m	conductivity of wires (S/m)
60.0	degrees	pitch angle of the braid (α)
4	integer	order of transfer impedance rational function model
log	log/lin	frequency range type for rational function model fitting
1E5 1E8 100	Hz Hz Integer	fmin, fmax, number of frequencies for rational function model fitting

Table 7.1 Cable braid parameters

The braid parameters are illustrated in figure 7.1

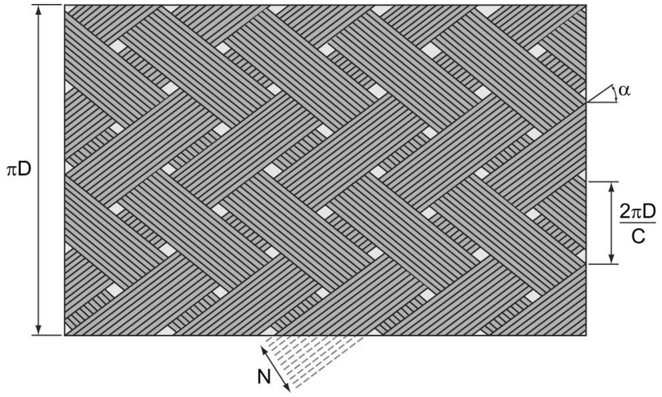


Figure 7.1 Braid parameters

The transfer impedance rational function representation is derived using the function function fitting process described in the theory manual, chapter 5. This process provides a best fit model of specified order over the specified frequency range. The model order can be specified in two ways:

1. The order is specified as a positive integer and this is the order used
2. A negative integer is specified. In this case the order is chosen using an automatic algorithm which attempts to choose the best order from 0 up to |specified order|

The frequency range for the model fit must be specified as must be the use of a log or linear frequency scale.

An example **.braid_spec** file is shown below

```
5E-3              ! braid diameter, D (m)
8                 ! Number of carriers, C
5                 ! Number of wires in a carrier, N
0.25e-3           ! diameter of a single wire, d (m)
5E7               ! conductivity of wires (S/m)
60.0              ! pitch angle of the braid (degrees)
4                 ! order of transfer impedance model
log               ! frequency range type
1E5   1E8  100    ! fmin, fmax, number of frequencies
```

To run the frequency dependent transfer impedance model generation process use the command:

shield_conductor_and_transfer_impedance_model_builder

The user is prompted to enter the name of the braid specification data (without **.braid_spec** extension). This is the only action required by the user.

The output of the process is a file **name.shield_model** which contains the parameters required to specify the shield model in a **.cable_spec** file i.e. the shield radius, the equivalent shield thickness, the shield conductivity and the rational function coefficients for the transfer impedance ratonal function model. In addition a rational function model for the shield surface resistance is given, plus some dervied values used in the calculation of the transfer impedance.

An example **.shield_model** file is shown below.

```
2.5000000000000001E-003  # Parameter Shield radius (m)
6.2500000000000015E-005  # Parameter Equivalent shield thickness (m)
50000000.000000000       # Parameter Shield conductivity (S/m)

# Transfer impedance model

628318530.71795857        # w normalisation constant
        6   # a order, a coefficients follow below:
2.0374795470322369E-002   2.4254670721262799        58.296168241008900
360.12236267295452        895.52398521190491        1000.0529344339834
485.68143255311600
        5   # b order, b coefficients follow below:
1.0000000000000000        23.599449123864325        145.90222072648217
```

```
362.78999275303266        405.14016972906620        196.75864271144215

# Conductor surface impedance model

628318530.71795857        # w normalisation constant
        4   # a order, a coefficients follow below:
2.0425431606526812E-002   1.4239191601879064        10.701214713988804
17.439082798989524        5.5415240356671571
        4   # b order, b coefficients follow below:
1.0000000000000000        18.987055251519244        64.911367837365901
9.386334031367127        4.7789439058672514

# Shield parameters used in the shield model calculation
braid circummference, cb=   1.5707963267948967E-002
C=           8  Number of carriers
N=           5  Number of conductors in each carrier
W=   1.2500000000000000E-003  Width of each carrier
W=   2.4999999999999996E-003  Width of each carrier in circumferential direction
cb/(C/2)=   3.9269908169872417E-003  circumferential dimension for each carrier
P=   441.06311633743348
v=   3528.5049306994679        Number of holes per unit length in braid
F=  0.63661977236758116        Fill factor
K=  0.86795481016581144        Optical coverage
l=   8.2387353231870868E-004   hole length
w=   1.4269908169872426E-003   hole width
S=   9.2336117727334644E-007   hole area
Th=   1.9846324228763292        hole attenuation factor
Ck=  0.12025223119877292        hole inductance factor
Ro   2.0371832715762598E-002   braid d.c. resistance
T    6.2500000000000015E-005   braid equivalent thickness
mean braid diameter, Dm=   5.4999999999999997E-003
Width between holes,  b=   7.1349540849362139E-004
Average height for braid inductance,  hh=   2.5000000000000001E-004
M12=(7.8743906839290131E-010,0.0000000000000000) Hole inductance
Mb =(3.1411609510809177E-009,0.0000000000000000) Braid inductance
Ms=(0.0000000000000000     ,0.0000000000000000) Skin inductance
M=(3.9286000194738192E-009,0.0000000000000000) Total transfer inductance
```

Chapter 8

Running the software with the GUI

This chapter should be read in conjunction with the other chapters of this user guide, it does not contain sufficient information to be read as a standalone document. As outlined in the previous chapter there are three processes involved in the creation of any Spice cable bundle model. These are:

1. cable model building process
2. cable bundle model building process
3. spice cable bundle model building process

This chapter will describe how the SACAMOS graphical user interface (GUI) is used to drive these processes as described in chapters 2 to 4, i.e.:

1. create a cable model
2. create a cable bundle model
3. create a Spice cable bundle model

Aiding the creation of all cable types. Namely:

1. Frequency Dependent Cylindrical conductor with dielectric
2. Frequency Dependent Coaxial cable with transfer impedance and shield surface impedance loss
3. Frequency Dependent Twinax cable with transfer impedance and shield surface impedance loss
4. Frequency Dependent Twisted pair

5. Frequency Dependent Shielded twisted pair with transfer impedance and shield surface impedance loss
6. Frequency Dependent Spacewire cable with transfer impedance and shield surface impedance loss
7. Frequency Dependent Overshield with transfer impedance and shield surface impedance loss
8. Frequency Dependent flex cable
9. D connector

Please Note: When using the GUI a strict directory structure must be adhered to. A parent SACAMOS folder must exist that follows the illustrative directory structure outlined in the figure below:

```
SACAMOS
|--BIN
|  `--All SACAMOS Executables (command line and GUI)
|--MOD1
|  |-- CABLE
|  |   |--single_wire.cable_spec
|  |   `--single_wire.cable
|  |-- BUNDLE
|  |   |--2_wire.bundle_spec
|  |   `--2_wire.bundle
|  `--SPICE
|     |--SYMBOL
|     |  |--2_wire.asy
|     |  `--2_wire.sym
|     |--2_wire.spice_model_spec
|     |--2_wire_Ngspice.lib
|     `--2_wire_LTspice.lib
|
|--MOD2
|  |-- CABLE
|  |-- BUNDLE
|  `-- SPICE
|      `-- SYMBOL
`--MOD3
   |-- CABLE
   |-- BUNDLE
   `-- SPICE
       `-- SYMBOL
```

In the file structure schematic more than one MOD folder is shown by way of illustrating the use of multiple cable model libraries. MOD1 shows the case for a; single wire cable model, bundle of 2 single wires, Spice model of 2 wire bundle and the associated library (.lib) and symbol files for Ngspice (.sym) and LTspice (.asy). MOD2 and MOD3 are shown empty, with only the required subfolder structure present.

8.1 Starting the GUI for the Creation of Spice Cable Bundle Models

The GUI for the creation of Spice cable bundle models is started by running the application 'SW1_GUI' which will open the main application window shown in figure 8.1 below:

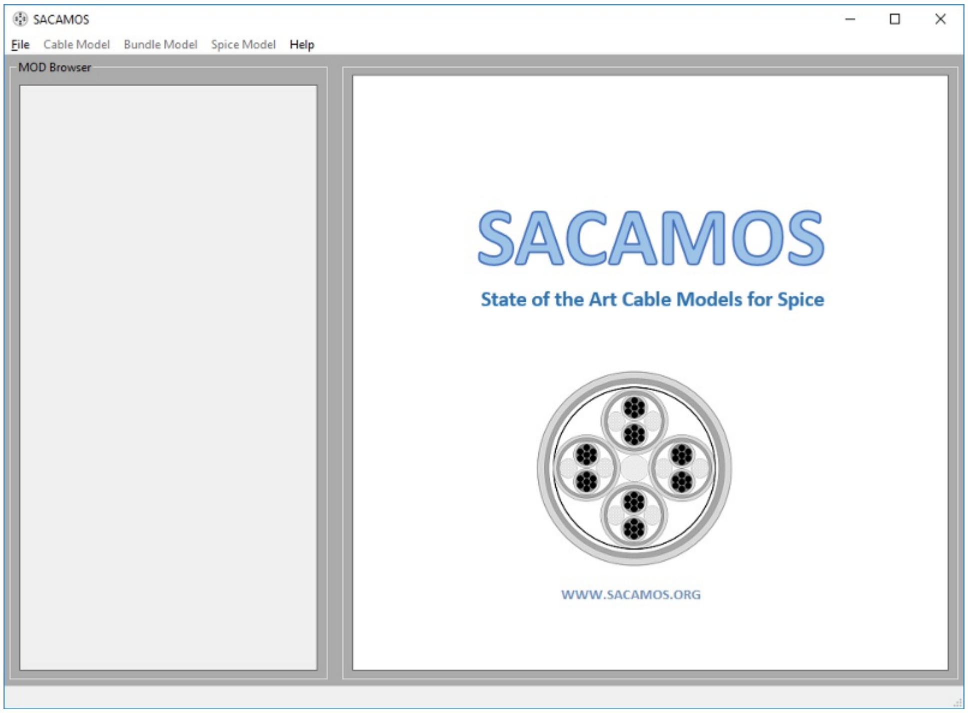


Figure 8.1 GUI main application window

With the exception of the File menu all items are initially disabled to force the selection or creation of a MOD library. Once selected (or created) the other menu items will become available - note the earlier stated constraint that any MOD library folder structure must sit at the same level as the executable (BIN) folder.

Throughout the following the example of an unshielded twisted pair cable will be used to illustrate the steps necessary to create a Spice model of a cable bundle.

8.1.1 Creation of the Cable Model

Select the required cable model type from the 'Cable Model' menu in the toolbar. In figure 8.2 a Shielded Twisted Pair model has been selected:

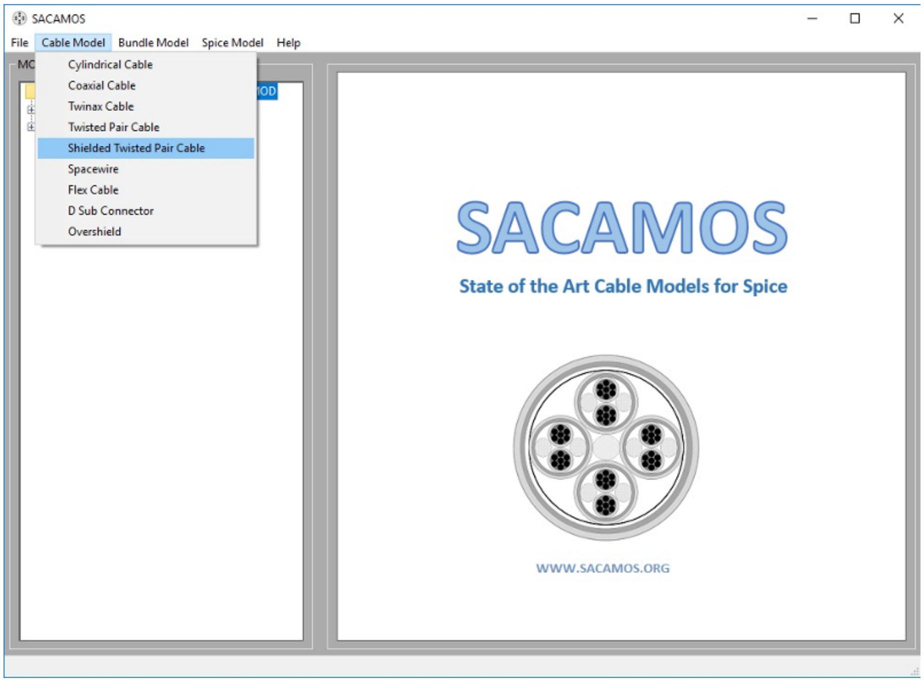


Figure 8.2 Cable Model Selection

This will open the cable designer form, figure 8.3. On opening, the Cable Name (1) and **general** properties (2) can be defined, as each property is selected it will be highlighted on the cable **cross section** (3). Once completed, clicking the **material** tab allows for the material properties to be defined, figure 8.4. For constant dielectric properties enter the required values (4), or for frequency dependent properties click the check box (5).

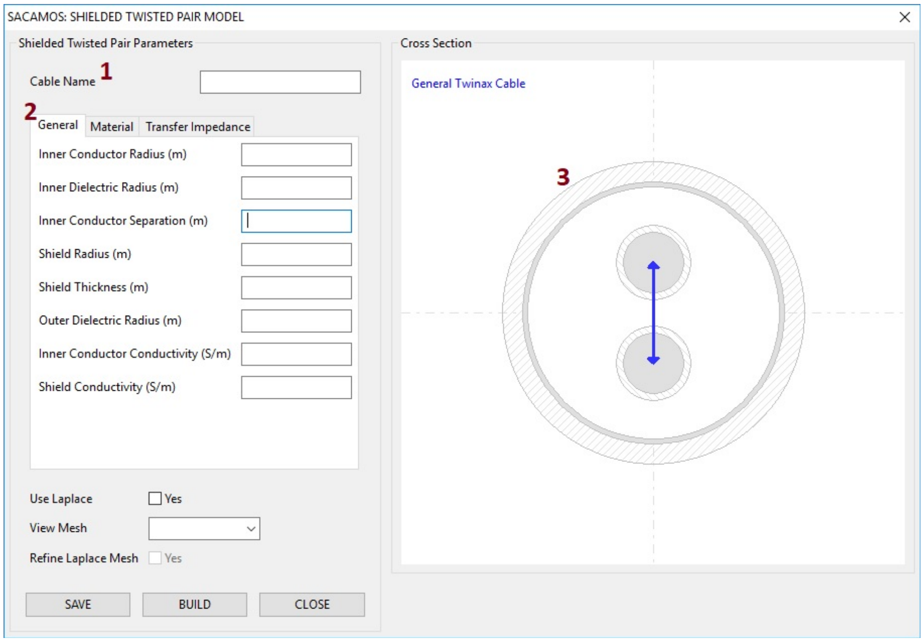

SACAMOS: SHIELDED TWISTED PAIR MODEL
Shielded Twisted Pair Parameters
Cable Name
1
2
General
Material
Transfer Impedance
Inner Conductor Radius (m)
Inner Dielectric Radius (m)
Inner Conductor Separation (m)
Shield Radius (m)
Shield Thickness (m)
Outer Dielectric Radius (m)
Inner Conductor Conductivity (S/m)
Shield Conductivity (S/m)
Use Laplace
□ Yes
View Mesh
Refine Laplace Mesh
□ Yes
SAVE
BUILD
CLOSE
Cross Section
General Twinax Cable
3

Figure 8.3 Cable Model Parameters, General Properties.

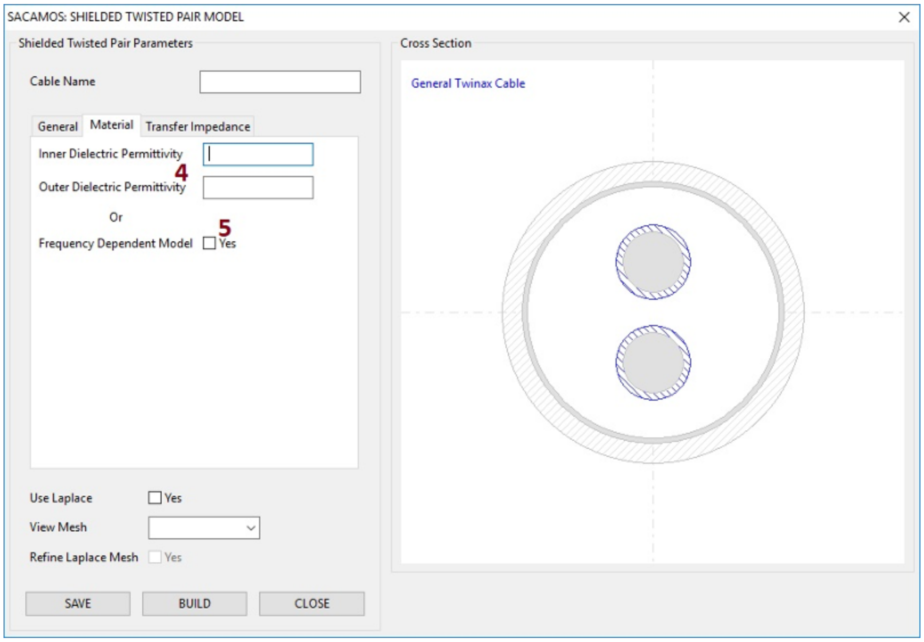
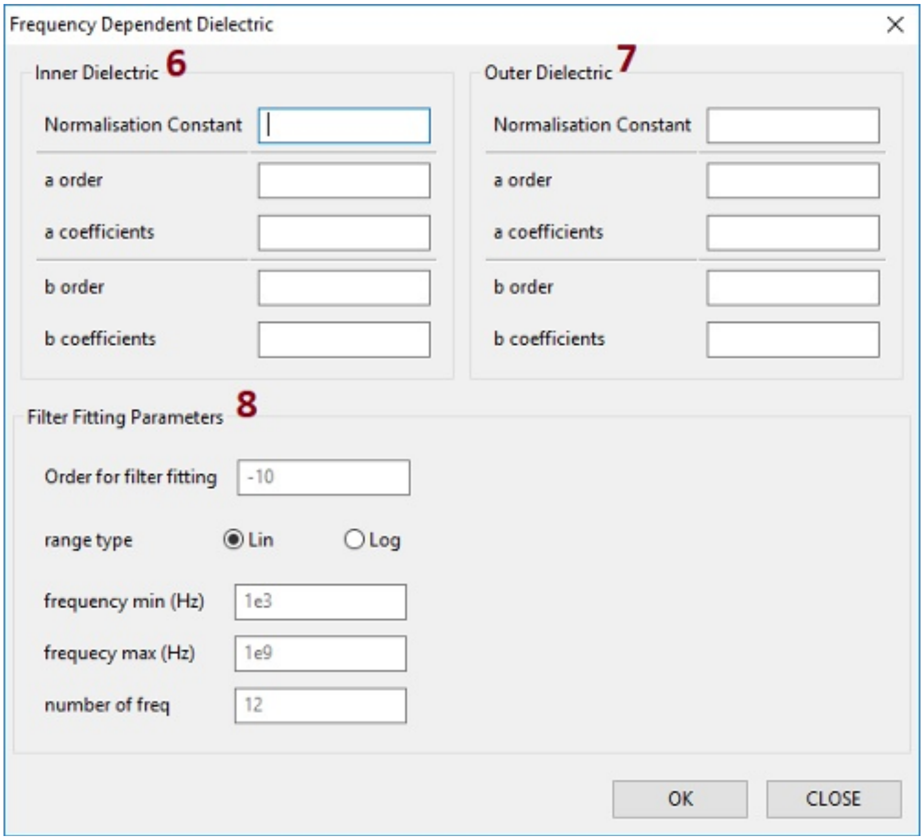

SACAMOS: SHIELDED TWISTED PAIR MODEL
Shielded Twisted Pair Parameters
Cable Name
General
Material
Transfer Impedance
Inner Dielectric Permittivity
4
Outer Dielectric Permittivity
Or
5
Frequency Dependent Model
□ Yes
Use Laplace
□ Yes
View Mesh
Refine Laplace Mesh
□ Yes
SAVE
BUILD
CLOSE
Cross Section
General Twinax Cable

Figure 8.4 Cable Model Parameters, Material Properties.

If **Frequency Dependent** properties are selected the dialogue box, figure 8.5 will be presented. Here the properties for the **inner dielectric** and **outer dielectric** materials, (6 & 7), can be set by specifying the rational functions that define the frequency dependence as described in chapter 2.2.1. The filter fitting parameters as described in chapter 2.2.2 are then set in (8), where the greyed values are suggested defaults. Note: not all cable types require all this information, for those types having only a single layer of dielectric or for those not requiring the filter fitting process the appropriate elements of the dialogue will be greyed out.



Inner Dielectric

Normalisation Constant

a order

a coefficients

b order

b coefficients

Outer Dielectric

Normalisation Constant

a order

a coefficients

b order

b coefficients

Filter Fitting Parameters

Order for filter fitting: -10

range type: ● Lin ○ Log

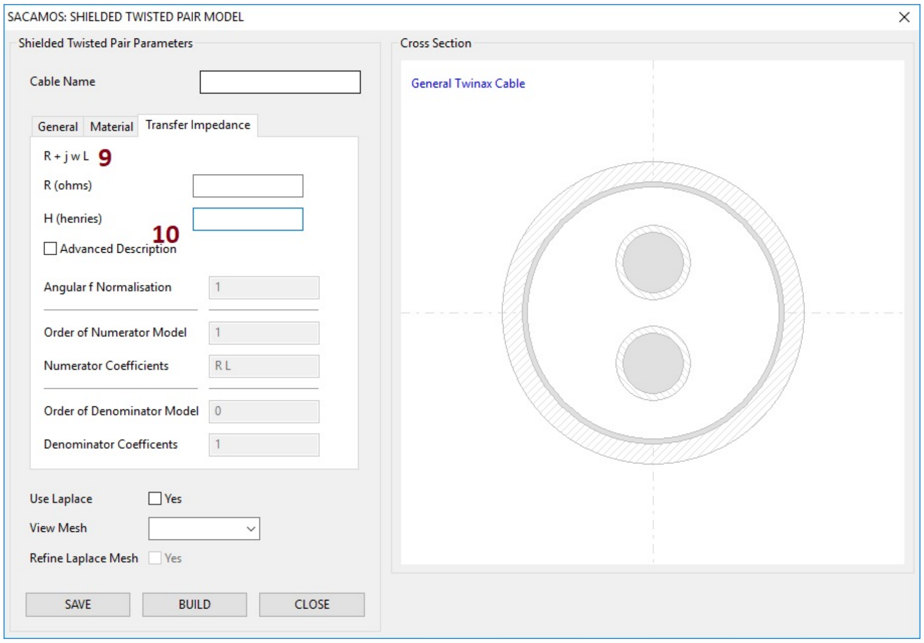
frequency min (Hz): 1e3

frequecy max (Hz): 1e9

number of freq: 12

Figure 8.5 Cable Model Parameters, Frequency Dependent Material Properties.

The **transfer impedance** tab allows the definition of the shield conductor transfer impedance which may be constant (9) or frequency dependent, figure 8.6. The latter enabled by selecting the check box (10). Further information on these properties can be found in chapter 2.2.1.



SACAMOS: SHIELDED TWISTED PAIR MODEL

Shielded Twisted Pair Parameters

Cable Name

R + j w L 9

R (ohms)

H (henries)

10

□ Advanced Description

Angular f Normalisation: 1

Order of Numerator Model: 1

Numerator Coefficients: R L

Order of Denominator Model: 0

Denominator Coefficents: 1

Use Laplace □ Yes

View Mesh

Refine Laplace Mesh □ Yes

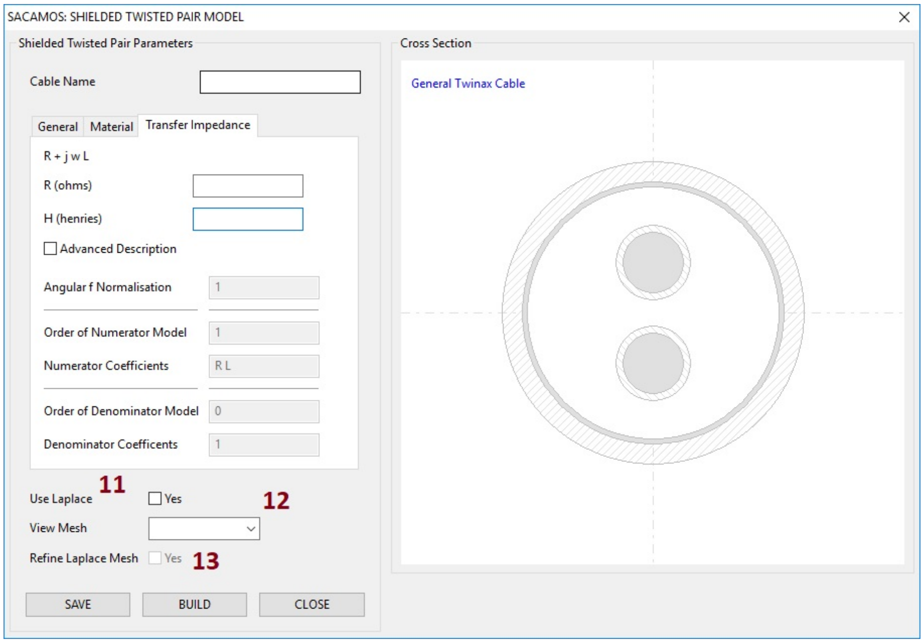
Cross Section

General Twinax Cable

Figure 8.6 Cable Model Parameters, Frequency Dependent Transfer Impedance Properties.

Finally the choice of whether to use an approximate analytic formula or the numerical Laplace solver to calculate the per unit length parameters is to be made, figure 8.7. Selecting **Use Laplace** (11) will invoke the Laplace solver and populate the mesh list (12). Selecting a mesh from this list will display the mesh used by the Laplace solver for a specific domain, a decision can then be made whether to **refine** the mesh (13) or accept the default mesh. If the mesh parameters are changed then the model must be saved again and the build process selected to re-build the model. A description of the mesh parameters can be found in chapter 2.2.2.

Once the cable has been completely specified, click **Save**. If the cable has been specified correctly click save will result in the button being greyed out, any errors will result in an appropriate error message indicating any incorrect or missing parameters. Next click build to generate the cable model which should result in a **Run Status** dialogue box, figure 8.8 indicating that the process has finished correctly. The cable model form can now be closed.



SACAMOS: SHIELDED TWISTED PAIR MODEL

Shielded Twisted Pair Parameters

Cable Name:

Transfer Impedance

R + j w L

R (ohms):

H (henries):

□ Advanced Description

Angular f Normalisation: 1

Order of Numerator Model: 1

Numerator Coefficients: R L

Order of Denominator Model: 0

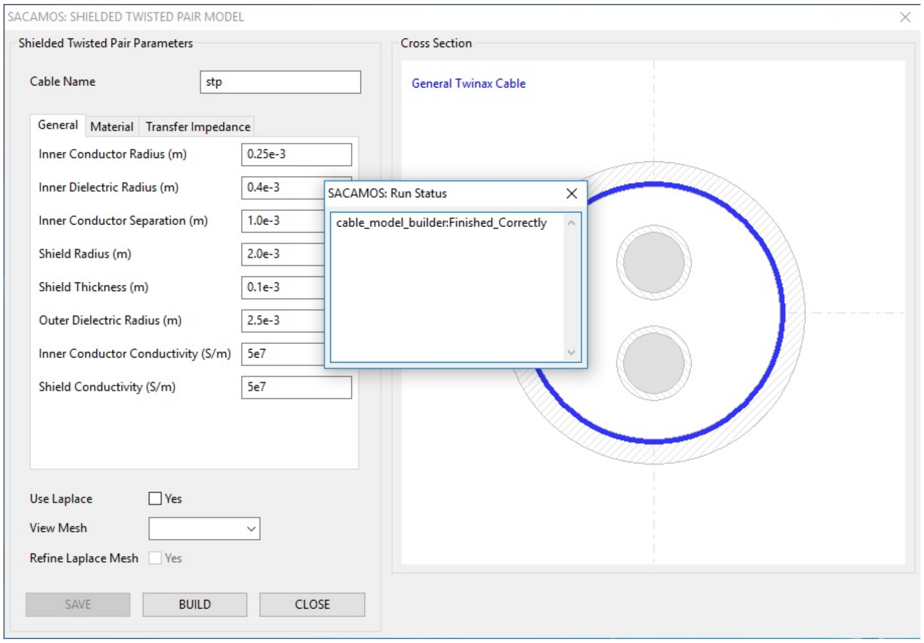
Denominator Coefficents: 1

Use Laplace □ Yes

View Mesh

Refine Laplace Mesh □ Yes

Figure 8.7 Cable Model Parameters, the Laplace Solver.



SACAMOS: SHIELDED TWISTED PAIR MODEL

Shielded Twisted Pair Parameters

Cable Name: stp

General

Inner Conductor Radius (m): 0.25e-3

Inner Dielectric Radius (m): 0.4e-3

Inner Conductor Separation (m): 1.0e-3

Shield Radius (m): 2.0e-3

Shield Thickness (m): 0.1e-3

Outer Dielectric Radius (m): 2.5e-3

Inner Conductor Conductivity (S/m): 5e7

Shield Conductivity (S/m): 5e7

Use Laplace □ Yes

View Mesh

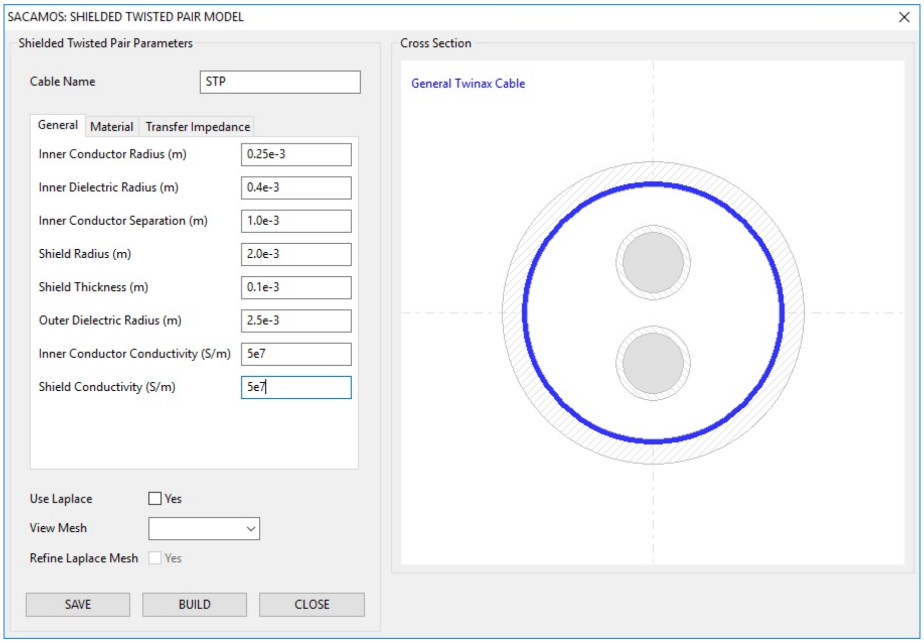
Refine Laplace Mesh □ Yes

SACAMOS: Run Status

cable_model_builder:Finished_Correctly

Figure 8.8 Cable Model Parameters, the Build Process Completed.

Figures 8.9, 8.10 and 8.11 show the completed forms for an example shielded twisted pair cable model, named **STP**.



Cable Name: STP

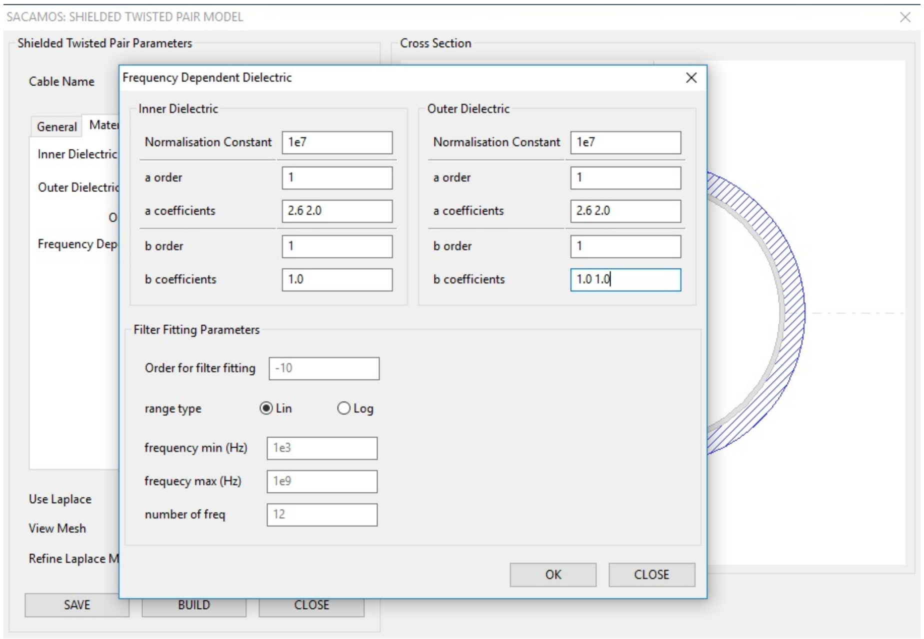
Inner Conductor Radius (m)	0.25e-3
Inner Dielectric Radius (m)	0.4e-3
Inner Conductor Separation (m)	1.0e-3
Shield Radius (m)	2.0e-3
Shield Thickness (m)	0.1e-3
Outer Dielectric Radius (m)	2.5e-3
Inner Conductor Conductivity (S/m)	5e7
Shield Conductivity (S/m)	5e7

Use Laplace □ Yes

View Mesh

Refine Laplace Mesh □ Yes

Figure 8.9 Cable Model Parameters: General Properties.



SACAMOS: SHIELDED TWISTED PAIR MODEL

Shielded Twisted Pair Parameters

Cable Name

General | Mater

Inner Dielectric

Outer Dielectric

O

Frequency Dep

Use Laplace

View Mesh

Refine Laplace M

SAVE | BUILD | CLOSE

Cross Section

Frequency Dependent Dielectric

Inner Dielectric

Normalisation Constant	1e7
a order	1
a coefficients	2.6 2.0
b order	1
b coefficients	1.0

Outer Dielectric

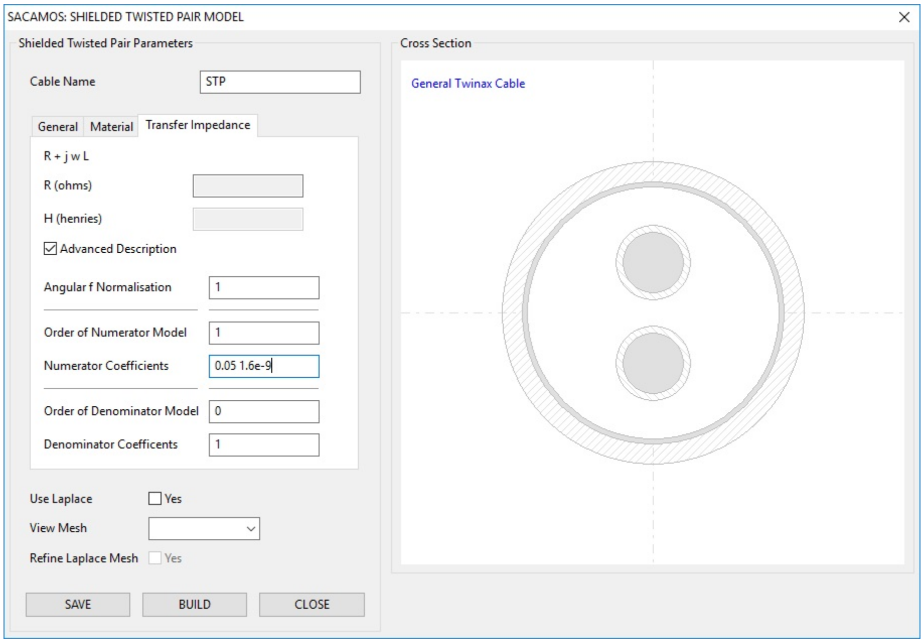
Normalisation Constant	1e7
a order	1
a coefficients	2.6 2.0
b order	1
b coefficients	1.0 1.0

Filter Fitting Parameters

Order for filter fitting	-10
range type	● Lin ○ Log
frequency min (Hz)	1e3
frequecy max (Hz)	1e9
number of freq	12

OK | CLOSE

Figure 8.10 Shielded Twisted Pair: Material Properties.



SACAMOS: SHIELDED TWISTED PAIR MODEL

Shielded Twisted Pair Parameters

Cable Name: STP

General | Material | Transfer Impedance

R + j w L

R (ohms)	
H (henries)	

☑ Advanced Description

Angular f Normalisation	1
Order of Numerator Model	1
Numerator Coefficients	0.05 1.6e-9
Order of Denominator Model	0
Denominator Coefficents	1

Use Laplace	□ Yes
View Mesh	
Refine Laplace Mesh	□ Yes

SAVE | BUILD | CLOSE

Cross Section

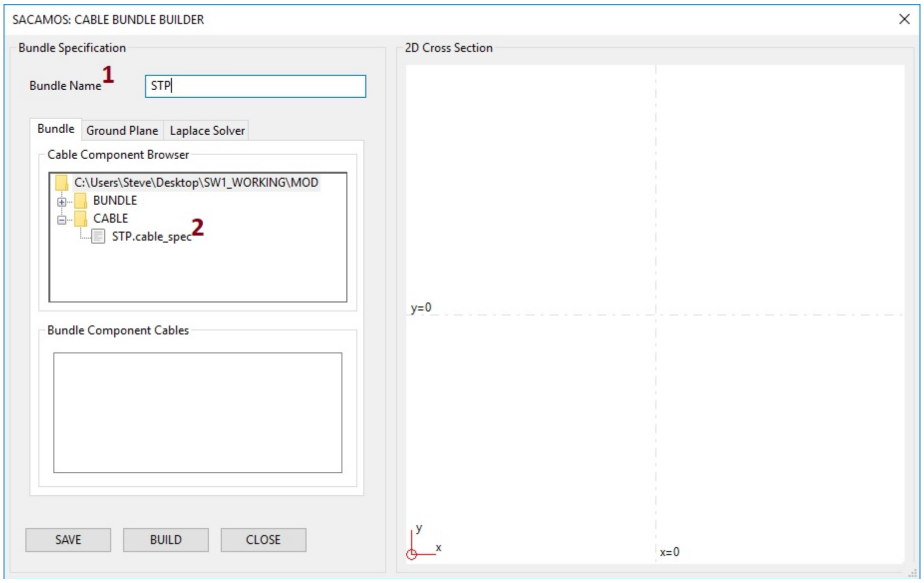
Figure 8.11 Shielded Twisted Pair: Transfer Impedance Properties.

The result is the creation of two files, STP.cable_spec and STP.cable, the specification and model files respectively, within the **CABLE** folder of the MOD libray.

```
|--MOD1
    |-- CABLE
    |    |--STP.cable_spec
    |    `--STP.cable
    |-- BUNDLE
    `-- SPICE
         `-- SYMBOL
```

8.1.2 Creation of the Cable Bundle Model

A cable bundle is built from constituent cable models that exist in MOD. To construct a bundle model, click **Bundle Model** from the menu bar and select Create Bundle to display the **Cable Bundle Builder** dialogue, figure 8.12. The **Bundle Name** can be specified here (1). In the **Cable Component Browser** all the available cables from which to build a bundle will be contained within the Cable folder. Click the + to expand the selection if necessary. The required cable components can be selected by double clicking them, opening a dialogue box with which to specify the location of the centre of the cable, figure 8.13.



SACAMOS: CABLE BUNDLE BUILDER

Bundle Specification

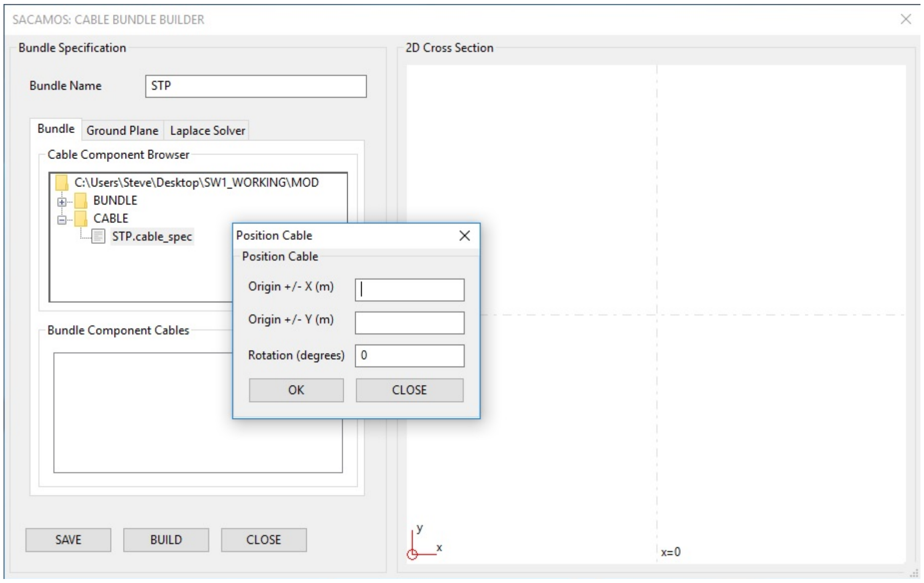
Bundle Name 1: STP

Cable Component Browser

C:\Users\Steve\Desktop\SW1_WORKING\MOD
- BUNDLE
- CABLE
  - STP.cable_spec 2

Figure 8.12 Bundle Model Builder: Selection of Component Cables

Once the position of the cable has been defined, clicking OK will add the cable to the **Component Cables** list (3) and show the cable cross section (4), figure 8.14.



SACAMOS: CABLE BUNDLE BUILDER

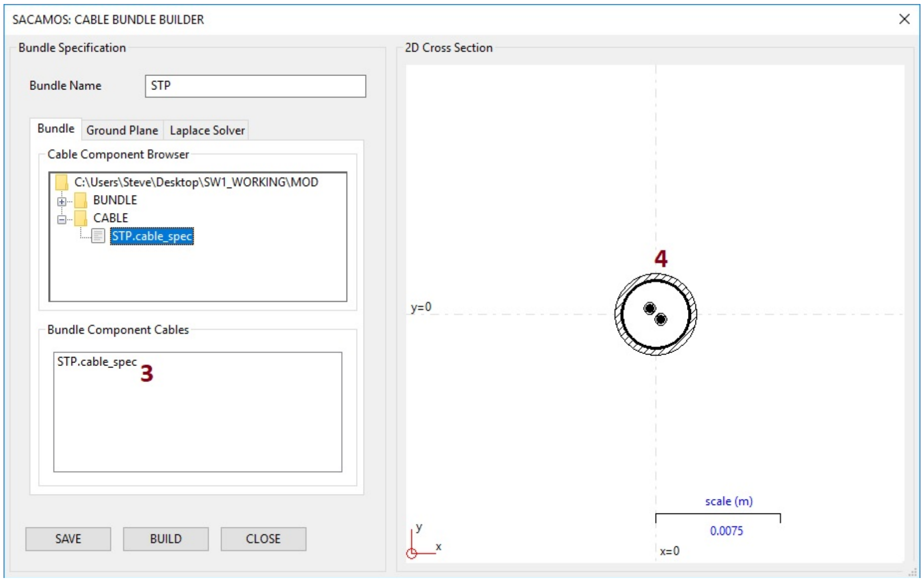
Bundle Name: STP

Origin +/- X (m):

Origin +/- Y (m):

Rotation (degrees): 0

Figure 8.13 Bundle Model Builder: Placement of Component Cables

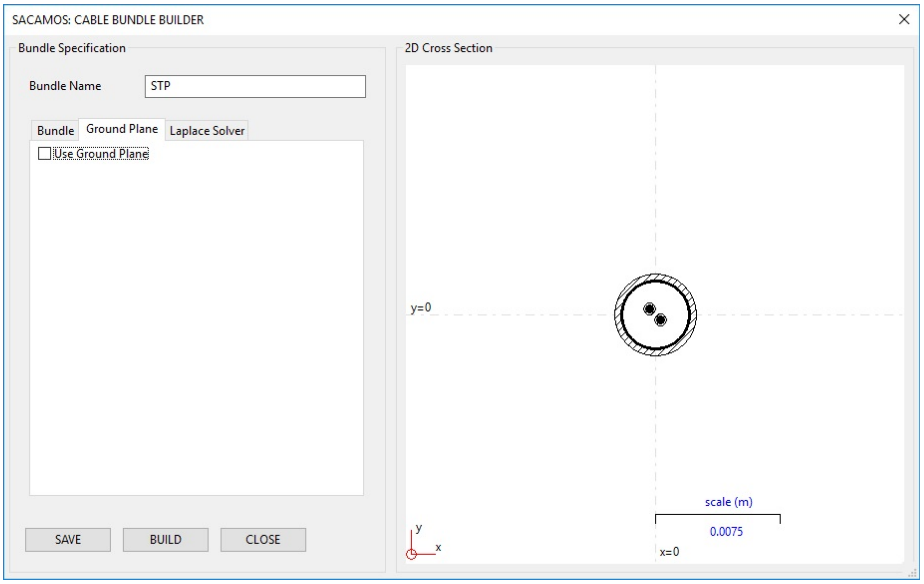


SACAMOS: CABLE BUNDLE BUILDER

Bundle Name: STP

Figure 8.14 Bundle Model Builder: Selection of Component Cables

From here further cables can be added to the bundle and/or a ground plane added, figure 8.15. Finally the use of the Laplace solver to calculate inductance and capacitance matrices for the external domain and any overshielded domains (by default, approximate analytic formulae are used), figure 8.16. See chapter 3.2 for further information. As previously described, if the Laplace solver is used the mesh list will be populated and the meshes can be viewed by selecting them from the drop down list and a decision then made whether to refine the mesh or accept the default.

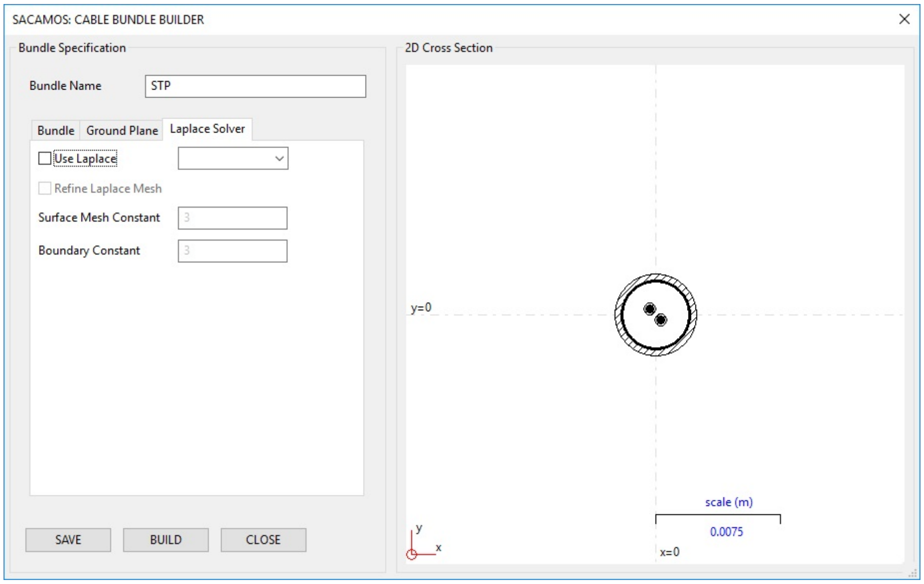

SACAMOS: CABLE BUNDLE BUILDER

Bundle Specification

Bundle Name: STP

□ Use Ground Plane

Figure 8.15 Bundle Model Builder: Ground Plane Inclusion



SACAMOS: CABLE BUNDLE BUILDER

Bundle Specification

Bundle Name: STP

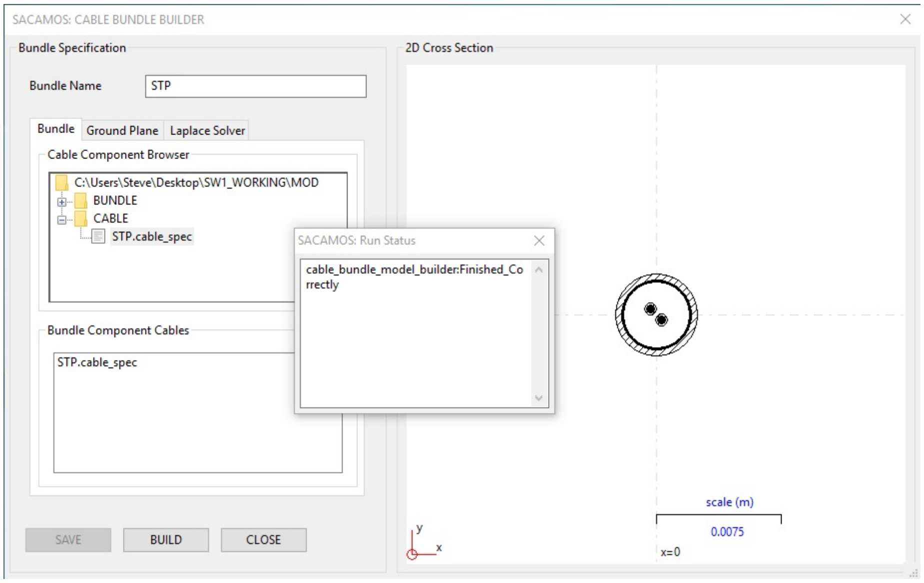
☐ Use Laplace

☐ Refine Laplace Mesh

Surface Mesh Constant: 3

Boundary Constant: 3

Figure 8.16 Bundle Model Builder: Laplace Solver



SACAMOS: CABLE BUNDLE BUILDER

Bundle Specification

Bundle Name: STP

Cable Component Browser

- C:\Users\Steve\Desktop\SW1_WORKING\MOD
  - BUNDLE
  - CABLE
    - STP.cable_spec

Bundle Component Cables

STP.cable_spec

SACAMOS: Run Status

```
cable_bundle_model_builder:Finished_Co
rrectly
```

Figure 8.17 Bundle Model Builder: Bundle Build Process Completed

Once the bundle has been completely specified, click **Save**. If the bundle has been specified correctly clicking save will result in the button being greyed out, any errors will result in an appropriate error message indicating any incorrect or missing parameters. Next click build to generate the bundle model which should result in a **Run Status** dialogue box, figure 8.17 indicating that the process has finished correctly. The cable model form can now be closed. The result is the creation of two files, STP.bundle_spec and STP.bundle within the **Bundle** folder of the MOD library, the specification and model files respectively.

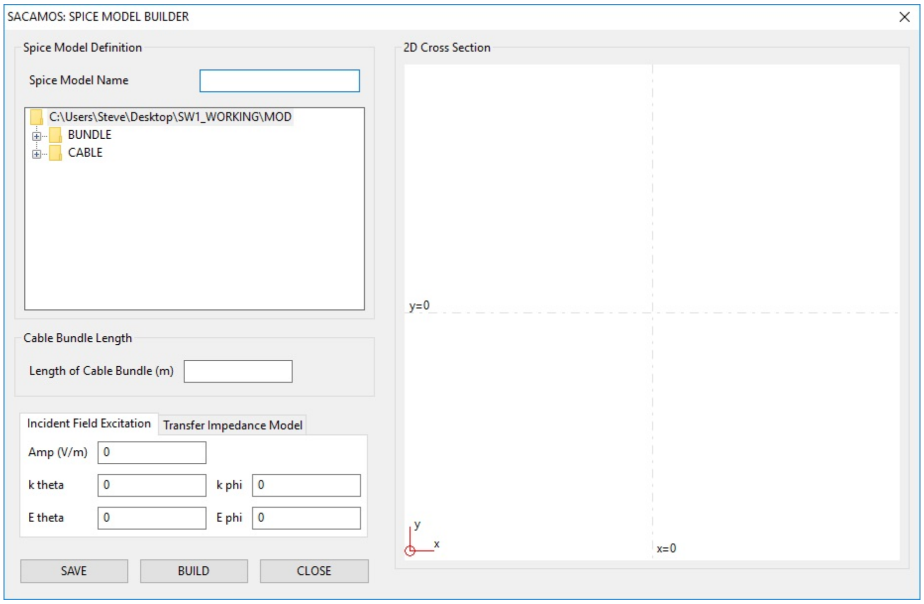
```
|--MOD1
   |-- CABLE
   |   |--STP.cable_spec
   |   `--STP.cable
   |-- BUNDLE
   |   |--STP.bundle_spec
   |   `--STP.bundle
   `-- SPICE
       `-- SYMBOL
```

8.1.3 Creation of the Spice Cable Bundle Model

A Spice cable bundle is built from a cable bundle model that exists in the MOD library. To construct a Spice cable bundle model, click **Spice Model** from the menu bar and select **Create Spice Bundle Model** to display the **Spice Model Builder** dialogue, figure 8.18. The **Spice Model Name** can be specified here. In the **Browser** all the available cables from which to build a Spice model will be contained within the Bundle folder. Click the + to expand the selection if necessary. The required bundle can be selected by double clicking the relevant bundle_spec file, displaying its cross section in the viewer, figure 8.19.

From here the length of cable represented by the Spice model is set (1). An incident field may also be defined here (2), see chapter 4 for further information on the definition of an incident field. In order to include a transfer impedance model (3) the conductor number of the shield of interest must be specified along with the direction of the coupling. The coupling direction is specified as either +1 or -1 where +1 indicates coupling direction from inside the shield to outside and -1 indicates coupling from the outside to the inside, further information can again be found in chapter 4.

Once the Spice model has been completely specified, click **Save**. If the Spice model has been specified correctly clicking save will result in the button being greyed out, any errors will result in an appropriate error message indicating any incorrect or missing parameters. Next click build to generate the Spice model which should result in a **Run Status** dialogue box, figure 8.20 indicating that the process has finished correctly. The Spice model form can now be closed. The result is the creation of the spice_model_spec file, two files defining the Spice library files (one each for LTspice and Ngspice) in the SPICE folder and their associated symbols in the SYMBOL FOLDER.



SACAMOS: SPICE MODEL BUILDER

Spice Model Definition

Spice Model Name

C:\Users\Steve\Desktop\SW1_WORKING\MOD
BUNDLE
CABLE

Cable Bundle Length

Length of Cable Bundle (m)

Incident Field Excitation | Transfer Impedance Model

Amp (V/m) 0

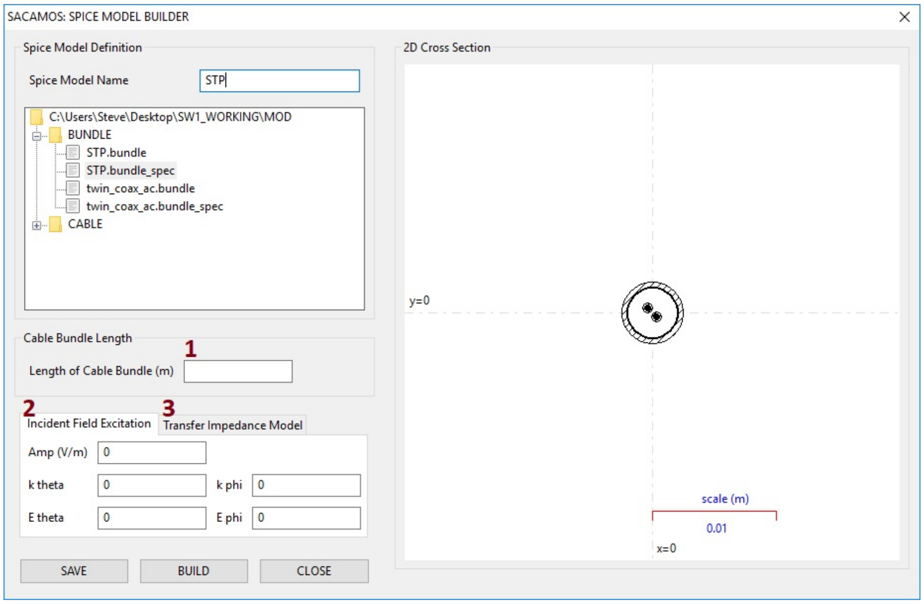
k theta 0 k phi 0

E theta 0 E phi 0

Figure 8.18 Spice Model Builder: Spice Model Specification Dialogue

```
|--MOD1
    |-- CABLE
    |    |--STP.cable_spec
    |    `--STP_wire.cable
    |-- BUNDLE
    |    |--STP.bundle_spec
    |    `--STP.bundle
    `-- SPICE
         |-- SYMBOL
         |    |--STP.asy
         |    `--STP.sym
         |-- STP.spice_model_spec
         |-- STP_Ngspice.lib
         `-- STP_LTspice.lib
```

The process for building all Spice bundle models is similar to that , only differing in the individual description of the different cable geometries.



SACAMOS: SPICE MODEL BUILDER

Spice Model Definition

Spice Model Name: STP

- C:\Users\Steve\Desktop\SW1_WORKING\MOD
  - BUNDLE
    - STP.bundle
    - STP.bundle_spec
    - twin_coax_ac.bundle
    - twin_coax_ac.bundle_spec
  - CABLE

Cable Bundle Length

Length of Cable Bundle (m):

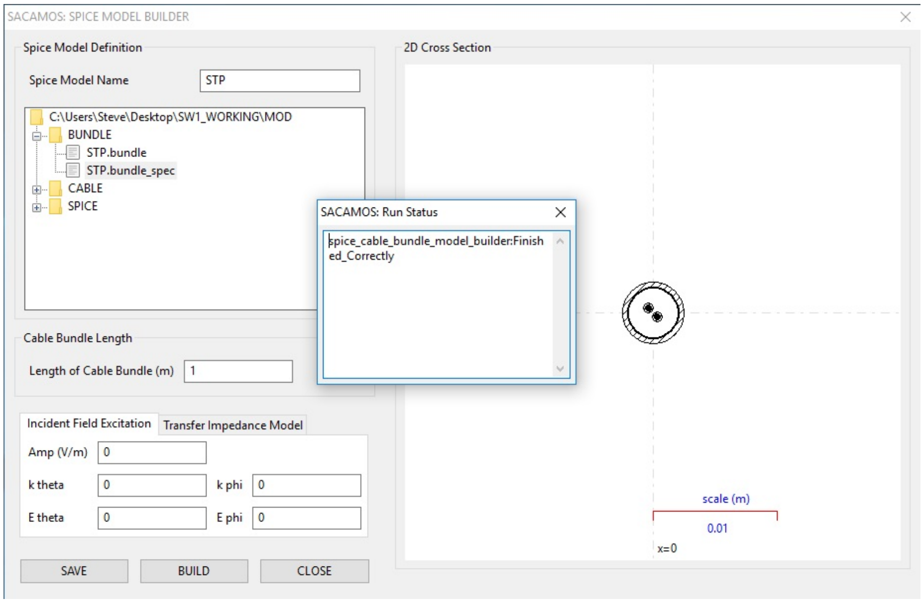
Incident Field Excitation

Amp (V/m): 0

k theta: 0 k phi: 0

E theta: 0 E phi: 0

Figure 8.19 Spice Model Builder: Spice Model Specification



SACAMOS: SPICE MODEL BUILDER

Spice Model Definition

Spice Model Name: STP

- C:\Users\Steve\Desktop\SW1_WORKING\MOD
  - BUNDLE
    - STP.bundle
    - STP.bundle_spec
  - CABLE
  - SPICE

Cable Bundle Length

Length of Cable Bundle (m): 1

Incident Field Excitation

Amp (V/m): 0

k theta: 0 k phi: 0

E theta: 0 E phi: 0

SACAMOS: Run Status

```
spice_cable_bundle_model_builder:Finish
ed_Correctly
```

Figure 8.20 Spice Model Builder: Spice Build Process Completed

Chapter 9

Using the Transmission Line models in Spice

This chapter describes how the models, once created, can be used in the different versions of Spice supported, namely Ngspice, LTspice and PSpice.

If the GUI is being used to generate the Spice cable bundle models then the path names indicated below may be set in the GUI so that the Spice subcircuit files and the associated symbol files are copied to the appropriate directories. If PSpice is being used then there is a step using the PSpice model editor which is required to set up the model and symbol for use in schematic capture.

9.1 Using Models in Ngspice

The Spice cable bundle model building process creates a subcircuit file **name.lib** and a corresponding symbol file **name.sym**.

For use by gshem the symbol file should be copied to the directory:

/usr/share/gEDA/sym/local

When runnning the Spice cable bundle model building process from the script **generate_spice_cable_bundle_model**, if you want to make the circuit symbols available for use by gshem you will need to add the following to the **generate_spice_cable_bundle_model** file:

SYMBOL_DIR="/usr/share/gEDA/sym/local"

Note also that you may need to change ownership of SYMBOL_DIR with something like the following command: sudo chown chris:chris /usr/share/gEDA/sym/local

The symbol file (**name.sym**) contains the path to the subcircuit library file for the transmission line model. The symbol file may require editing to ensure that the path is correct on any given system. The line which must be checked is

file=**PATH_TO_TRANSMISSION_LINE_SUB_CIRCUITS/name.lib**

9.2 Using Models in LTspice

The Spice cable bundle model building process creates a subcircuit file **name.lib** and a corresponding symbol file **name.asy**.

In order to use the subcircuit and the associated symbol in LTspice running on windows, the symbol file, **name.asy**, should be copied into the LTspice symbol directory or a sub-directory created within this directory.

C:\Program Files\LTC\LTspiceV\lib\sym

The symbol file contains the path to the subcircuit library file for the transmission line model. The symbol file may require editing to ensure that the path is correct on any given system. The line which must be checked is

SYMATTR ModelFile **C:\PATH_TO_TRANSMISSION_LINE_SUB_CIRCUITS\name.lib**

One this is set up correctly the symbol can be chosen and included in the schematic like any other though it is important to note that LTspice must be re-started before any changes to the symbol library take effect.

9.2.1 Using Models in PSpice

The Spice cable bundle model building process creates a subcircuit file **name.lib**. When using this model in PSpice there is an initial stage of creating a Cadence PSpice library file which includes an associated symbol. The process required is as follows:

1. Start the PSpice model editor.

2. Load the subcircuit file (**name.lib**) using the model input wizard (file → Model Input Wizard). The dialog box automatically fills the name of the library file to be produced (**name.olb**)

3. Reply 'Yes' to the question 'Do you want to attach the default rectangular symbol'. At this point the library file **name.olb** is produced.

4. If you would like the model to be available in the default libraries loaded then you will need to edit the file: **C:\Cadence\SPB_17.2\tools\PSpice\library\normd.lib** to include the line

   .lib "name.lib"

   If this step is not done then the library file will have to loaded explicitly by editing the 'Simulation Settings' in Orcad Capture, before running the simulation.

   The library will have to be loaded when placing parts in the schematic.

9.3 Spice 'node zero'

Care may be required in specifying node zero in Spice circuits containing transmission line models. Figure 9.1 shows a typical circuit configuration in which two termination sub-circuits are connected via a three conductor transmission line.

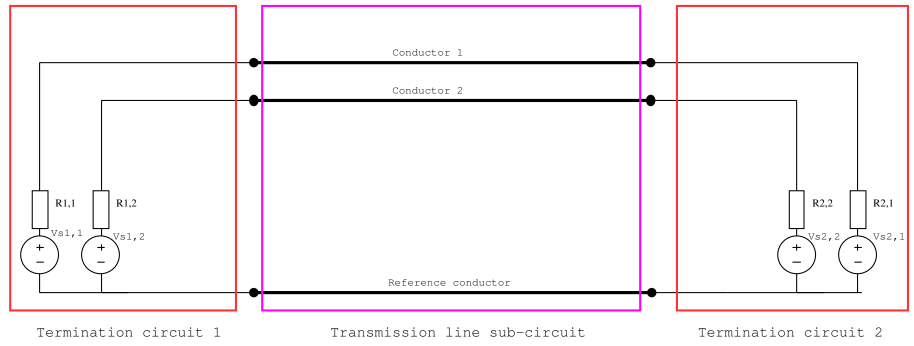


Figure 9.1 Example circuit using a transmission line sub-circuit model

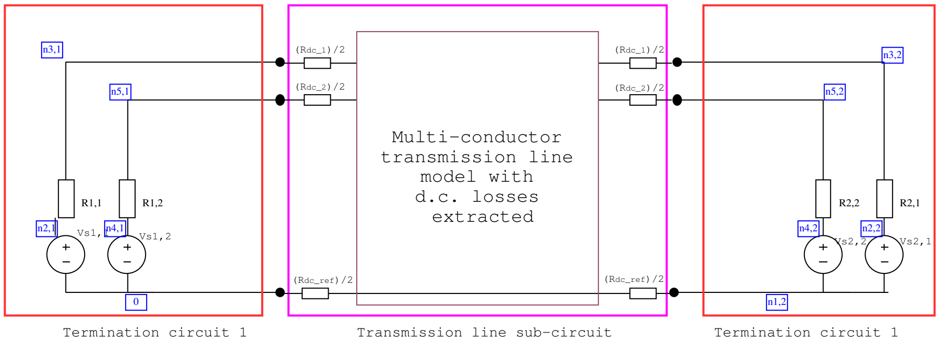


Figure 9.2 Example circuit with node zero in one termination circuit only

Figure 9.2 shows a depiction of a Spice model of this configuration in which termination 1 includes node zero and termination 2 does not include node zero. This circuit will work with the transmission line model operating as it is designed to, even if there is some loss in the reference conductor. It is important to note that voltages in termination circuit 1 should be measured relative to node zero however voltages in termination circuit 2 should be measured relative to the local reference node, in this case node $n1, 2$.

Figure 9.3 shows a depiction of a Spice model in which both termination 1 and termination 2 include node zero as a reference. In this case, the d.c. resistance, R_{dc_ref}, of the reference conductor is effectively short circuited as depicted by the heavy line in figure 9.3 and the transmission line model may not give accurate results, especially if the d.c. resistance is significant and the analysis is at low frequency. If the reference conductor is a ground plane (which is assumed to be perfectly conducting in this work) then having node zero at both ends of the transmission line sub-circuit will not affect the results (since $R_{dc_ref} = 0$ in this case). I can also be observed that an EMC oriented system model will include bonding resistances and inductances between the electronic units at either end and the ground plane. These bonding impedances will most tend to dominate the

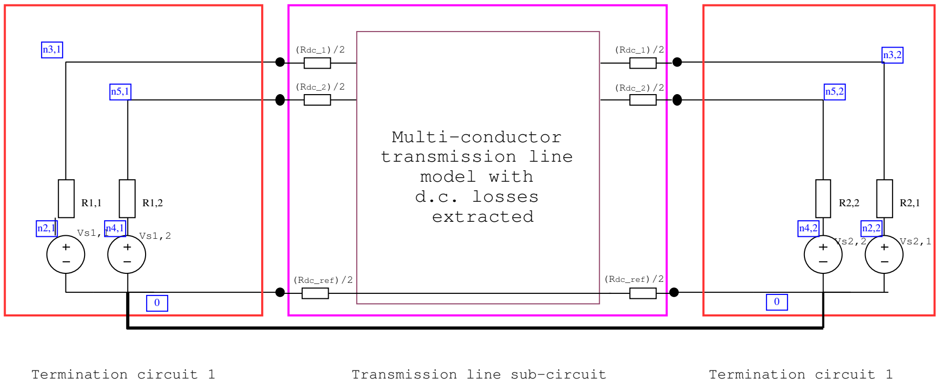


Figure 9.3 Example circuit with node zero in both termination circuits showing how node zero at both ends will short out any resistance in the reference conductor

d.c. resistances of the cables and thus the effect of shorting out the d.c. resistance of the reference conductor may have a small impact.

To conclude, we recommend that node zero is only included in one transmission line termination circuit and that all voltages in a transmission line termination are measured relative to the local termination reference node. Node zero should only be present at both ends of a transmission line sub-circuit if there is no loss in the reference conductor.

9.4 Non-convergence issues in Spice and possible solutions

It is not uncommon for Spice simulations to fail to run to completion due to non-convergence of the solution to the specified tolerances. This is a complicated issue due to the different solution algorithms which are used in each of the Spice analysis options. The main convergence issues are related to the following:

(a) Newton Raphson algorithm for non-linear equations

(b) D.C. bias point calculation

(c) Transient Analysis

(d) Timestep control

These aspects of Spice and the options available to change the Spice operation to improve convergence are discussed for each. Note that not all options are available in all versions of Spice. A detailed discussion of these issues is found in reference [9]

Newton Raphson algorithm

The circuit equations are solved for non-linear circuit elements using the Newton Raphson algorithm. The iteration stops when the following conditions are satisfied:

1. For all the nodes in the circuit: |V(t)-V(t-dt)| < RELTOL*V(t)+VNTOL

2 For all the branch currents in the circuit: |I(t)-I(t-dt)| < RELTOL*I(t)+ABSTOL

RELTOL is a relative tolerance whose default value is 0.001

VNTOL has a default value of 1μV

ABSTOL has a default value of 1pA

If there is a convergence error in the Newton Raphson solution then it is worth considering increasing VNTOL and/or ABSTOL. VNTOL can be set to a value RELTOL*Vsmall where Vsmall is the smallest voltage of interest. Similarly, ABSTOL can be set to a value RELTOL*Ismall where Ismall is the smallest current of interest

Convergence may also be improved by increasing GMIN. This is the minimum conductance which is allowed between nodes. GMIN should be set to be as large as possible without affecting the circuit operation i.e. work out the smallest resistance, R which can be connected between the circuit nodes then set GMIN=1/R.

If non-linear devices are included in the circuit then convergence may be improved by ensuring that non-linear models include some series resistance in the .MODEL definition.

D.C. bias point calculation

If Spice fails to find a d.c. bias point then the following actions may help:

1. Increase ITL1, the number of d.c. bias point iterations from the default of ITL1=100 to ITL1=500 (say)

2. Try to assist the d.c. convergence by setting some initial node values for the convergence calculation e.g. .NODESET V(n)=X

3. Use source stepping by setting ITL6 to 500 for example

Transient Analysis

In order to aid transient analysis, all non-linear device models should have their associated capacitance values set to a non-zero value.

The number of iterations for convergence at a time point can also be increased for example set ITL4=40 (default value is 10). If this value is exceeded then the timestep is reduced.

Timestep control

Automatic timestep control can aid convergence and speed of the solution however there are some issues to be aware of.

1. The timestep is always less than the print interval (TSTEP) so this can be used to set a limit on the timestep.

2. Increase ITL4 to prevent the timestep being reduced too readily. This may improve the total simulation time. The default value is 10.

Bibliography

[1] C.R. Paul, *Analysis of Multi-conductor Transmission Lines*, Second edition, John Wiley & Sons, 2008, ISBN 978-0-470-13154-1.

[2] H. Xie, J. Wang, R. Fan, Y. Liu, *SPICE Models to Analyze Radiated and Conducted Susceptibilities of Shielded Coaxial Cables*, IEEE Trans EMC, Vol 52, No 1, pp 215-222, 2010.

[3] Christopher Smartt, David Thomas, Steve Greedy, Jaco Verpoorte, Jesper Lansink Rotgerink and Harmen Schippers *Theory Manual, SACAMOS: State of the Art CAble MOdels for Spice. Open Souce Cable Models for EMI Simulations*

[4] B.K.P. Scaife, *Principles of Dielectrics*, Oxford University Press, 1998

[5] S. A. Schelkunoff *The Electromagnetic Theory of Coaxial Transmission Lines and Cylindrical Shields* Bell System Technical journal, Vol 13, No 4, 1934, pp 532-579.

[6] E. F. Vance *Shielding effectiveness of braided wire shields*, IEEE transactions on EMC, vol EMC-17, no 2, 1975, pp71-77

[7] T. Kley *Optimized single-braided cable shields* IEEE Transactions on Electromagnetic Compatibility, vol.35, no.1, pp.1,9, Feb 1993

[8] F. M. Tesche, M. V. Ianov, T. Karlsson, *EMC Analysis Methods and Computational Models* John Wiley and Sons, 1997.

[9] R. Kielkowski *Inside Spice*, Mc Graw Hil, 1998.

Appendix A

Transfer Impedance model according to Kley

From the work of Vance [6] and Kley [7] and the summary of this work in [8] we may calculate the frequency dependent transfer impedance of a braided wire shield from the cable shield parameters. The shield parameters are

i. braid diameter, D
ii. Number of carriers, C
iii. Number of wires in a carrier, N
iv. diameter of a single wire, d
v. conductivity of wires, σ
vi. pitch angle of the braid, α

These parameters are illustrated in figure A.1.

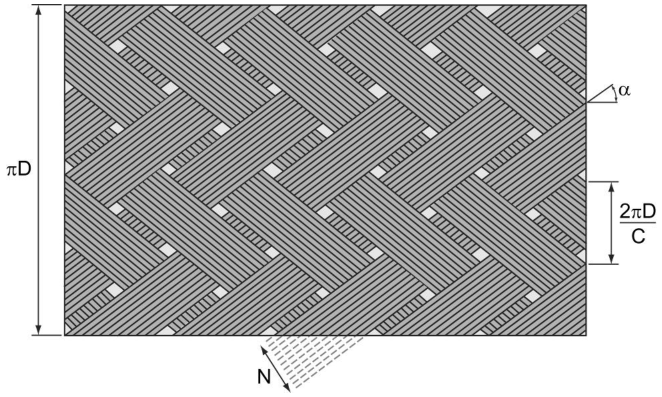


Figure A.1 Braid parameters

The transfer impedance is the sum of the effects of diffusion through the shield conductor, penetration of the field through the holes in the shield and the effects of the overalpping weave of the braid conductors.

$$Z_t = Z_d + j\omega M_h + j\omega M_b \tag{A.1}$$

The calculation of the transfer impedance terms proceeds as follows:
The fill factor, F, is calculated as

$$F = \frac{NCd}{2\pi D cos(\alpha)} \tag{A.2}$$

The optical coverage, K, is

$$K = 2F - F^2 \tag{A.3}$$

The length and width of the rhombic holes are given by

$$l = \frac{(1-F)\,Nd}{F sin(\alpha)} \tag{A.4}$$

$$w = \frac{(1-F)\,Nd}{F cos(\alpha)} \tag{A.5}$$

a parameter, e, is defined as

$$e = \sqrt{1 - \left(\frac{w}{l}\right)^2} \tag{A.6}$$

The diffusion impedance term is given by

$$Z_d = R_0 \frac{\gamma}{sinh(\gamma t)} \tag{A.7}$$

where R_0 is the d.c. resistance of the shield

$$R_0 = \frac{4}{\pi d^2 NC\sigma cos(\alpha)} \tag{A.8}$$

γ is the propgation constant in the shield conductor

$$\gamma = \frac{1+j}{\delta} \tag{A.9}$$

where

$$\delta = \sqrt{\frac{2}{\omega\mu_0\sigma}} \tag{A.10}$$

and t is an equivalent thickness of the cable shield given by

$$t = \frac{1}{\pi\sigma D R_0} \tag{A.11}$$

The hole inductance term is given by

$$M_h = \frac{\mu_0 \pi (1-K)^{\frac{3}{2}})e^2}{6C(Em(e)-(1-e^2)Km(e))} * C_k \tag{A.12}$$

where Km and Em are elliptic integrals and

$$C_k = 0.875e^{-T_h} \tag{A.13}$$

$$T_h = 9.6F\left(\frac{K^2 d}{2a}\right)^{\frac{1}{3}} \tag{A.14}$$

$$a = D/2d0 + d \tag{A.15}$$

$$M_b = \frac{-0.22\mu_0 d}{4\pi D_m F cos(\alpha)} cos(2k_1\alpha) \tag{A.16}$$

where

$$k_1 = \frac{\pi}{2.667F cos(\alpha) + \frac{\pi}{10}} \tag{A.17}$$

and

$$D_m = D + 2d \tag{A.18}$$