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: August 2, 2017
DOCUMENT VERSION: 1.1

---

[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). A bundle may consist of multiple individual cables usually with the addition of a ground plane (satellite panel). 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 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 theortical derivation of the spice multi-conductor transmission line models is described in [1].

Validation of the models is described in the accompanying document [2].

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

1. Ngspice
2. LTspice
3. Pspice

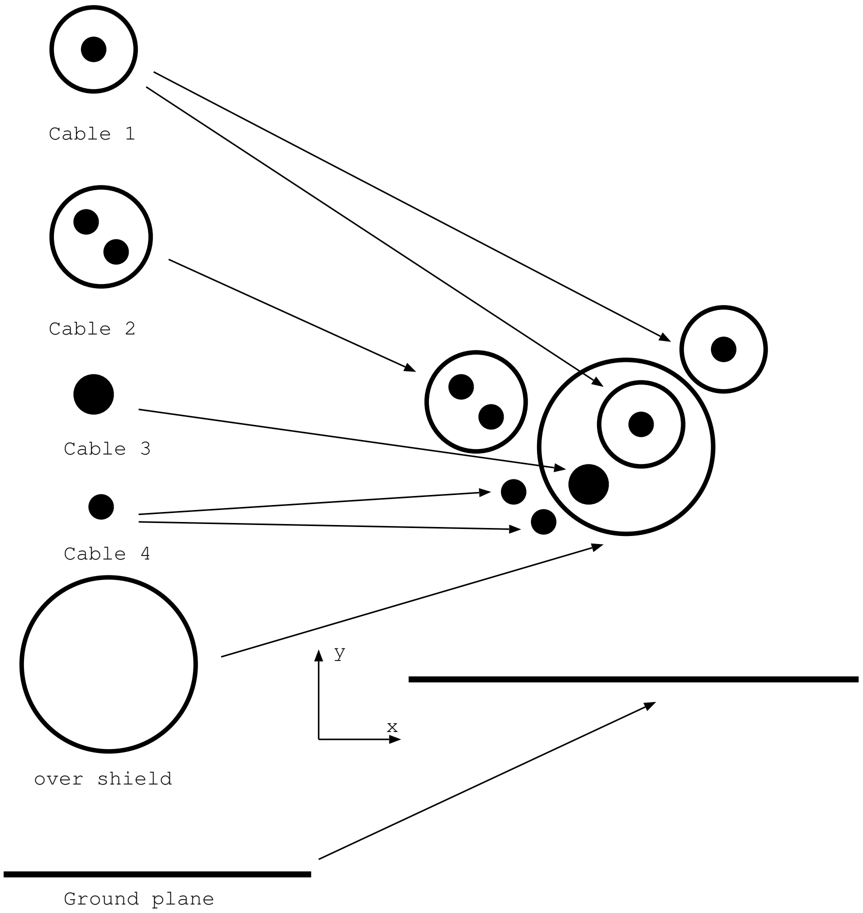


Figure 1.1 A complex cable bundle built up from individual cables, over-shield and ground plane

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 in 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.

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.

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).

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 the different cable type 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 paramters (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. 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 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

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.

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.

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

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

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

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 [3]. 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 commmon approximation in which the imaginary part of permittivity is assumed to be coonstant (constant $tan\left(\delta\right)$ model) is unphysical.

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. 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 [4]. 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 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.

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. Cable models with frequency dependent transfer impedance models are:

1. Frequency Dependent Coaxial cable with transfer impedance and shield surface impedance loss
2. Frequency Dependent Twinax cable with transfer impedance and shield surface impedance loss
3. Frequency Dependent Shielded twisted pair with transfer impedance and shield surface impedance loss
4. Frequency Dependent Spacewire with transfer impedance and shield surface impedance loss
5. Frequency Dependent 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. 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 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 inputs required are 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. 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 buildeing process.

An example of a filter fitting specification is shown below:

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

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.

If the Laplace solver is used then the mesh generation is be controlled by the paramter '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 **.bundle_spec file**:

```
Laplace_surface_mesh_constant
5
```

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 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 the previous section, 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 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 paramters is described in table 2.1 follwed by an example.

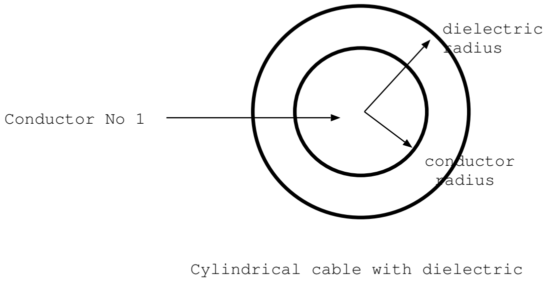


Figure 2.1 Cylindrical conductor with dielectric

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

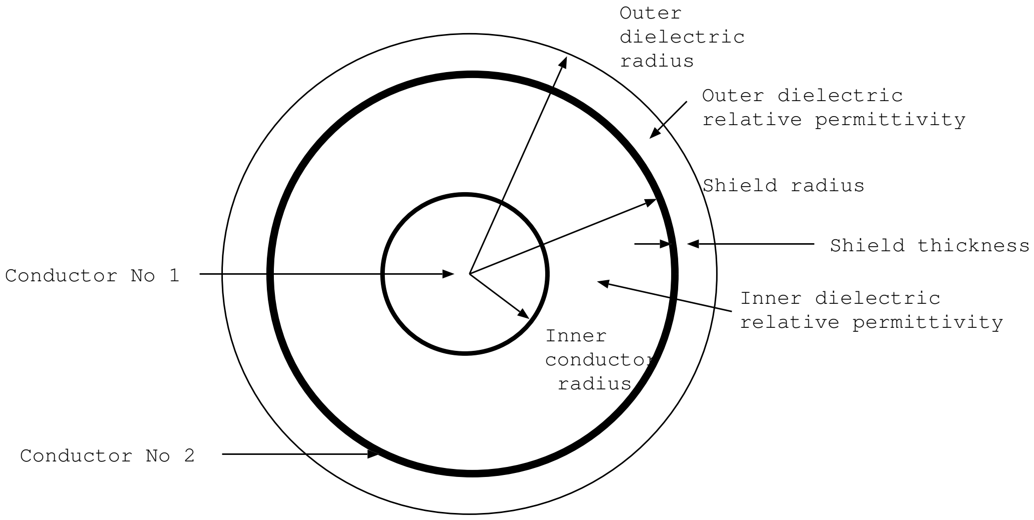
Figure 2.2 shows the cross section of the frequency dependent coaxial cable with transfer impedance and surface impedance loss. A description of the cable paramters is described in table **??** follwed by an example.

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

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

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

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

```
#MOD_cable_lib_dir
LIBRARY_OF_CABLE_MODELS
Coax
```

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

```
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 paramters is described in table **??** follwed by an example.

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.

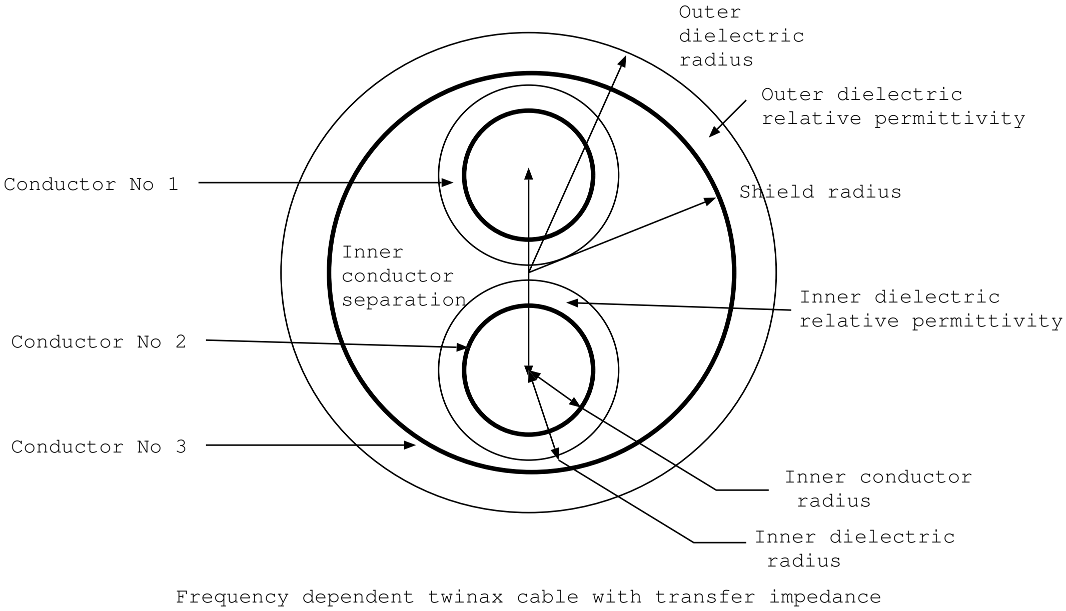


Figure 2.3 Twinaxial cable

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
```

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

```
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...
```

Frequency Dependent Twisted pair cable

Figure 2.4 shows the cross section of the frequency dependent twisted pair model. A description of the cable paramters is described in table 2.4 follwed 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

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

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

where the dielectric coating of the conductors is neglected.

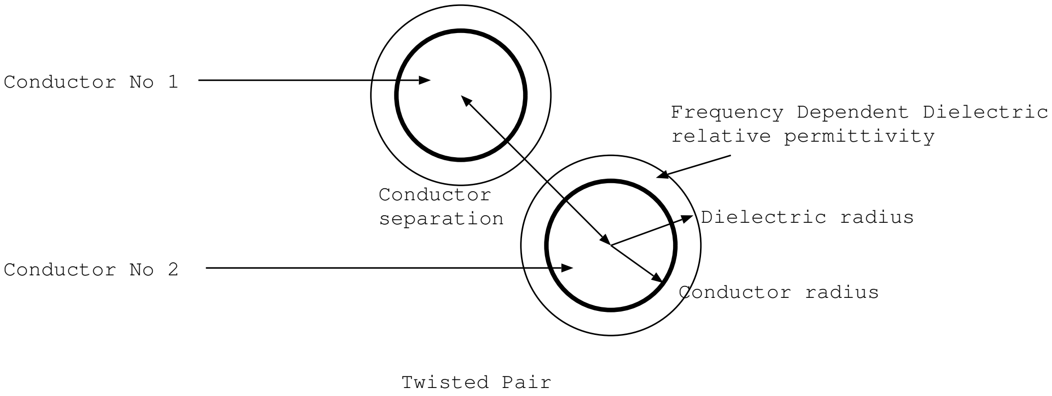


Figure 2.4 Twisted pair cable

Example

```
#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
```

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

```
 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 paramters is described in table 2.5 follwed by an example. 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.

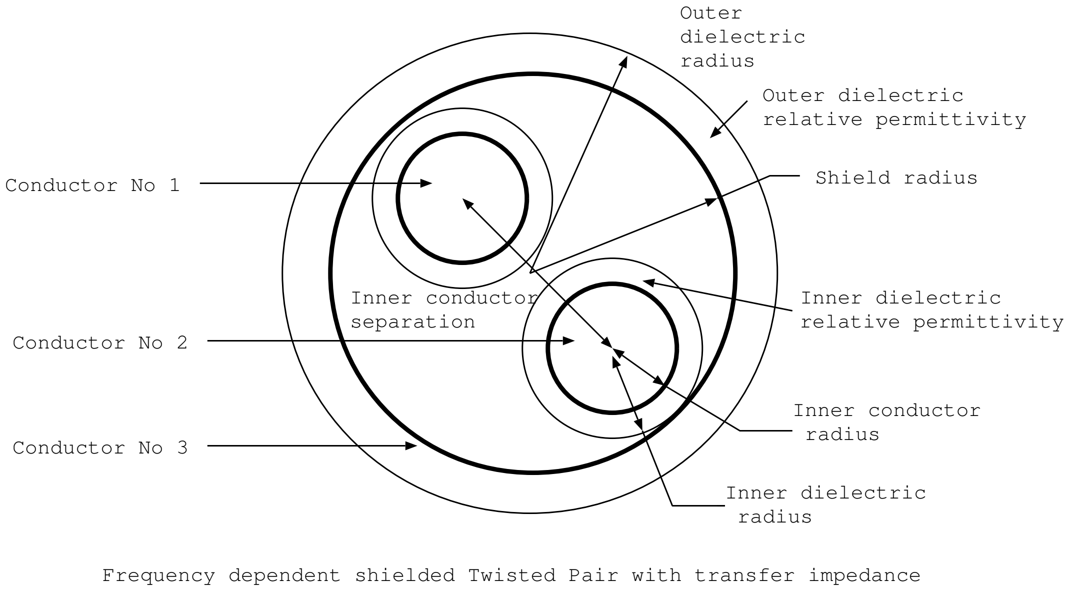


Figure 2.5 Shielded twisted pair

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
```

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
$\epsilon(j\omega)$	rational function coefficients	Frequency dependent inner dielectric relative permittivity model
$\epsilon(j\omega)$	rational function coefficients	Frequency dependent outer dielectric relative permittivity model
1	integer	number of frequency dependent transfer impedance models
$Z_T(j\omega)$	ohms (rational function coefficients)	Frequency dependent transfer impedance model

Table 2.5 Shielded twisted pair parameters

```
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...
```

Frequency dependent spacewire 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 paramters is described in table 2.6 follwed by an example. 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.

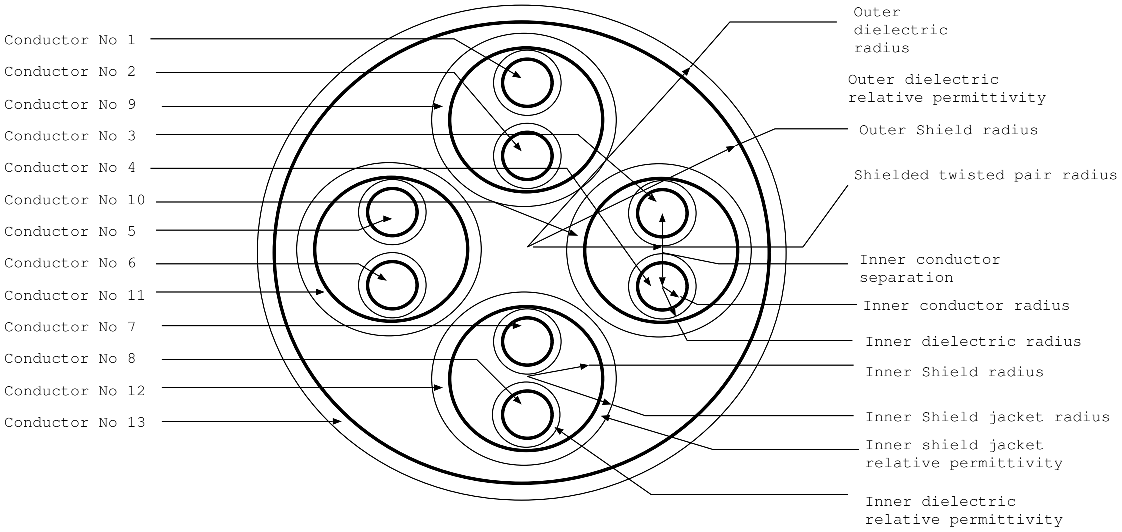


Figure 2.6 Spacewire

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...
```

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

Over Shield

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

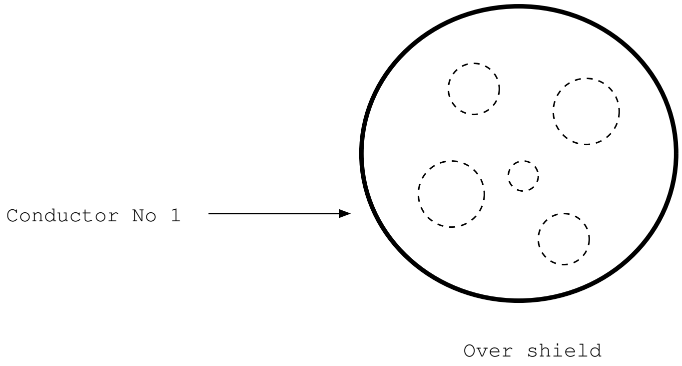


Figure 2.7 Over shield

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
$Z_T(j\omega)$	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

Figure 2.8 shows the cross section of the flex cable model. A description of the model paramters is described in table 2.8 follwed 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.

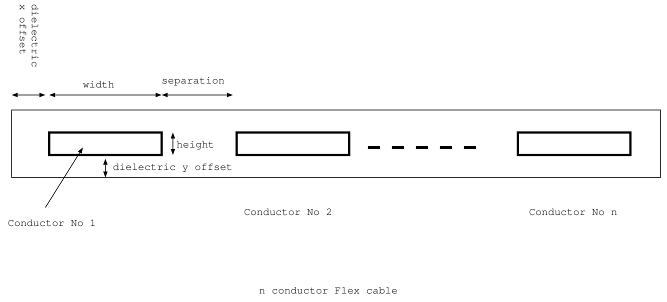


Figure 2.8 flex cable

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, always 6 for flex cables
1.0e-3	metre	parameter 1: conductor width (x)
0.2e-3	metre	parameter 2: conductor height (y)
0.6e-3	metre	parameter 3: conductor separation (x)
1.0e-3	metre	parameter 4: dielectric offset in x
0.2e-3	metre	parameter 5: dielectric offset in y
5E7	Siemens/metre	parameter 6: conductivity
1	integer	number of frequency dependent parameters
$\epsilon(j\omega)$	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
```

```
6 # number of parameters
1.0e-3   # parameter 1: conductor width (x dimension)
0.25e-3   # parameter 2: conductor height (y dimension)
0.5e-3   # parameter 3: conductor separation (x dimension)
0.2e-3   # parameter 4: dielectric offset x
0.1e-3   # parameter 5: dielectric offset y
5E7           # parameter 6: conductivity
1 # number of frequency dependent parameters
# dielectric relative permittivity model follows
   1E9          # w normalisation constant
           1    # a order, a coefficients follow below:
  2.2      2.0
           1    # b order, b coefficients follow below:
  1.0      1.0
```

D connector

Figure 2.9 shows the cross section of the Dconnector model. A description of the model paramters is described in table 2.9 follwed by an example. The Dconnector 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.

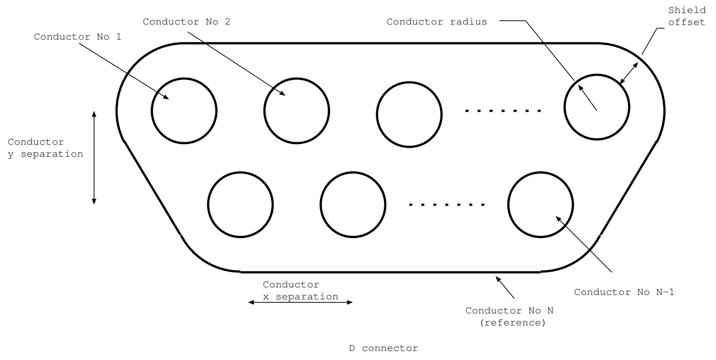


Figure 2.9 D connector

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
$\epsilon\left(j\omega\right)$	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 along the x axis.

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.

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 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.

The final part of the bundle specification concerns the specification of a ground plane (if required).

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 and choice of the reference conductor to be the last conductor in the bundle has no impact on the results from the model.

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 inputs required are the cables which constitute the bundle and their configuration in the bundle cross section, an indication of the presence or absence of a ground plane and if a ground plane is present, its configuration in the bundle cross section.

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.

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

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 ground plane point), bundle_size. The outer boundary is defined as a circle of radius $bundle_{s}ize*Laplace_{b}oundary_{c}onstant$. 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 $\frac{r}{Laplace_{s}urface_{m}esh_{c}onstant}$. The default value is 3.

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
```

A ground plane may be included in the bundle as shown in figure 3.1

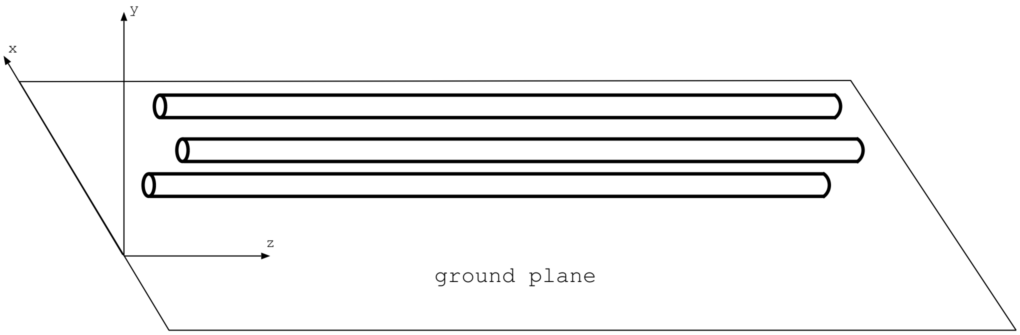


Figure 3.1 Specification of the ground plane position in the bundle cross section

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 to
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 bundlle

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
6.35e-4     -0.001  0.0
single_wire
6.35e-4     0.001  0.0
ground_plane
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 4.1.

4.0.1 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)

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

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. the polarisation vector is normalised to unit length and only the relative amplitudes of

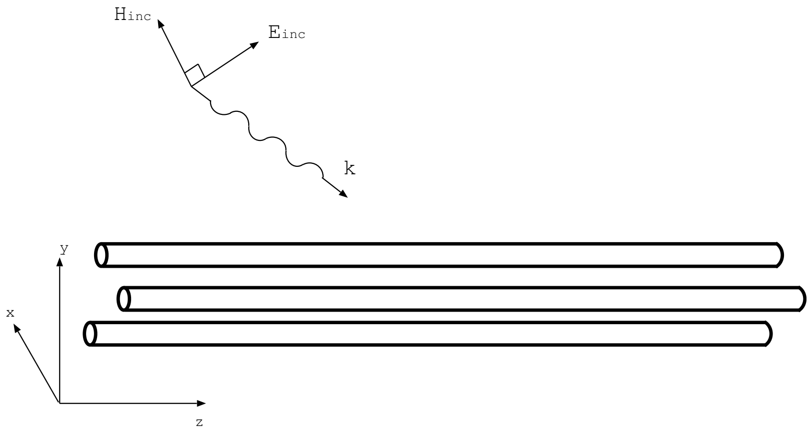


Figure 4.1 Plane wave incident field excitation of a cable bundle

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 $k_\theta = 0^\circ$ $k_\phi = 0^\circ$ $E_\theta = 1$ $E_\phi = 0$

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

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

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

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

4.0.2 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.

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

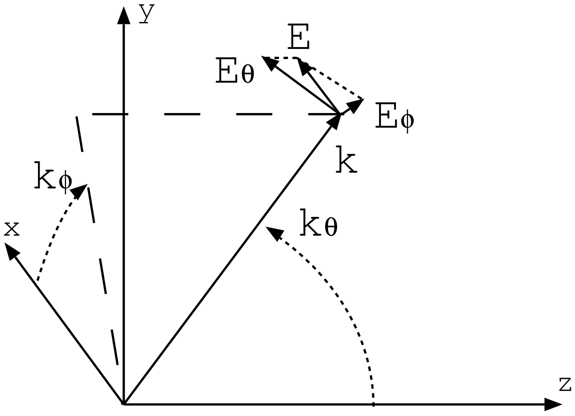


Figure 4.2 Specification of the incident field polarisation

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 the specification of transfer impedance coupling within a spice cable bundle model is seen in section4.1

4.0.3 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.

Spice 'node zero'

Care may be required in specifying node zero in spice circuits containing transmission line models. Figure 4.3 shows a typical circuit configuration in which two termination sub-circuits are connected via a three conductor transmission line.

Figure 4.4 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

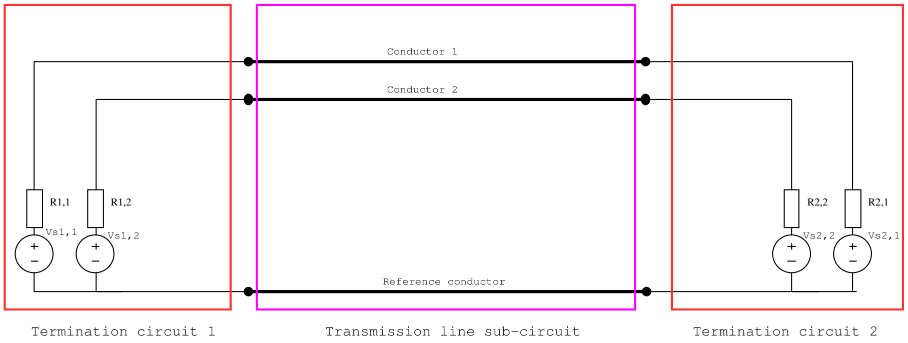


Figure 4.3 Example circuit using a transmission line sub-circuit model

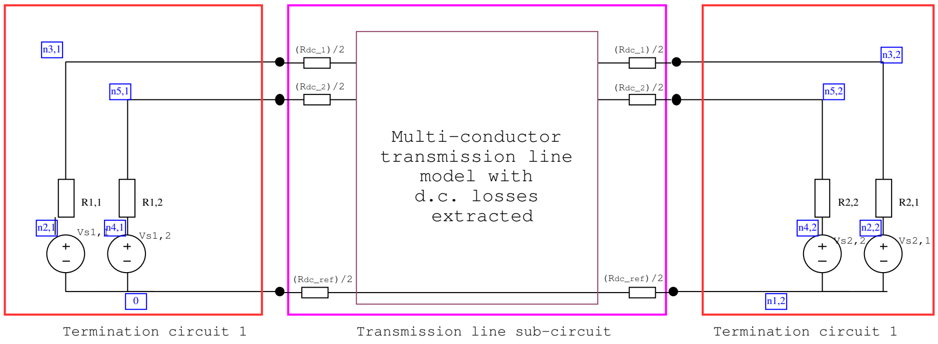


Figure 4.4 Example circuit with node zero in one termination circuit only

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 4.5 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 4.5 and the transmission line model will not give accurate results. If the reference conductor has no loss (a perfect ground plane for example) 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).

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.

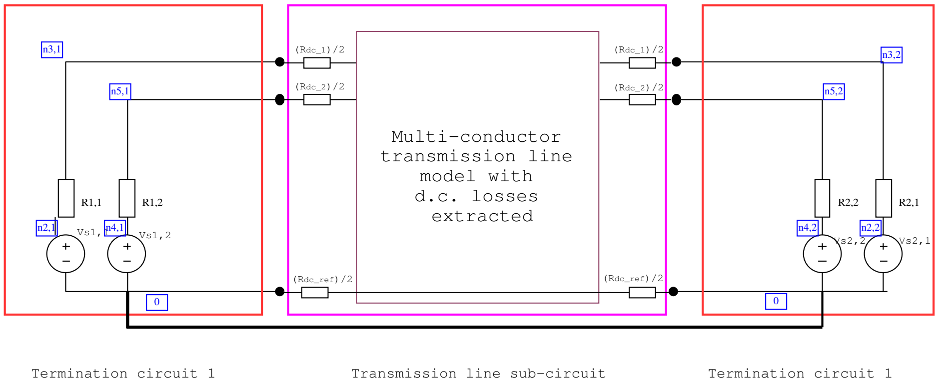


Figure 4.5 Example circuit with node zero in both termination circuits showing how node zero at both ends will short out any resistance in the reference conductor

4.1 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 **name.spice_model_spec**.

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

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 4.6

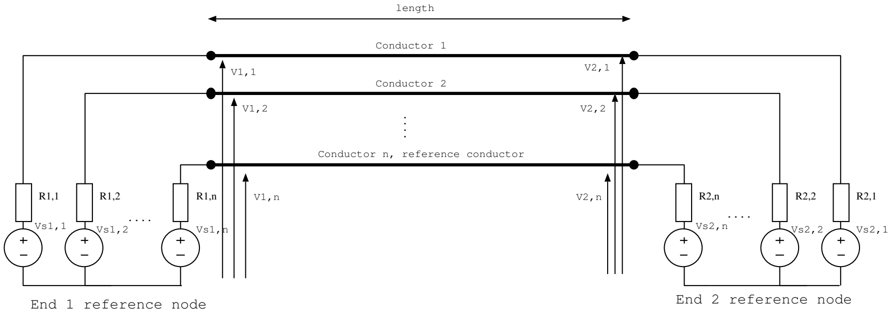


Figure 4.6 Validation test case configuration

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 re-

sults.

2. 'use_xie' use Xie's model for incident field excitation of shielded cables.
3. 'no_s_xfer' For first order frequency dependent models (transfer impedance, propagation correction) we may use a passive circuit implementation for frequency dependent transfer functions instead of s-domain transfer functions. This may be of use in Ngspice models which fail to run.

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 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	degrees	Polarisation Eθ Eϕ
15	# End 1 termination model	-	Number of sources and resistances = number of conductors-1
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
-	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

AC example

```
#MOD_cable_lib_dir
LIBRARY_OF_CABLE_MODELS
#MOD_bundle_lib_dir
LIBRARY_OF_BUNDLE_MODELS
#MOD_spice_bundle_lib_dir #LIBRARY_OF_BUNDLE_MODELS/two_wires_over_ground
./
#spice_symbol_dir
SYMBOL_DIR
# Specification for spice model of single wire over ground, no loss
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

```
#MOD_cable_lib_dir
.
```

```
#MOD_bundle_lib_dir
.
MOD_spice_bundle_lib_dir
.
#spice_symbol_dir
SYMBOL_DIR
2   # number of conductors, n. Conductor n is always the reference conductor
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 impdeance 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 —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
-	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
-	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 resis-

AC example for a frequency dependent coaxial cable with a transfer impedance model plus a single wire

```
#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 1 +
# 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
```

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
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
5. 'no_high_freq_zt_model' (default) 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
6. '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 experimental and only available for 1st and 2nd order models at the moment.

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_MODELS**) 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 cantains all the information required to characterise a particular bundle configuration. Cable bundle models are kept in the directory **BUNDLE_MODELS**. 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 source and victim conductors etc.) Within the **BUNDLE_MODELS** directory a sub-directory with the same name as the cable bundle model is created. This sub-directory contains all the Spice cable bundle models developed for the bundle.

An example of the library of cable models generated in the running of the test cases is shown in figure 5.1 below.

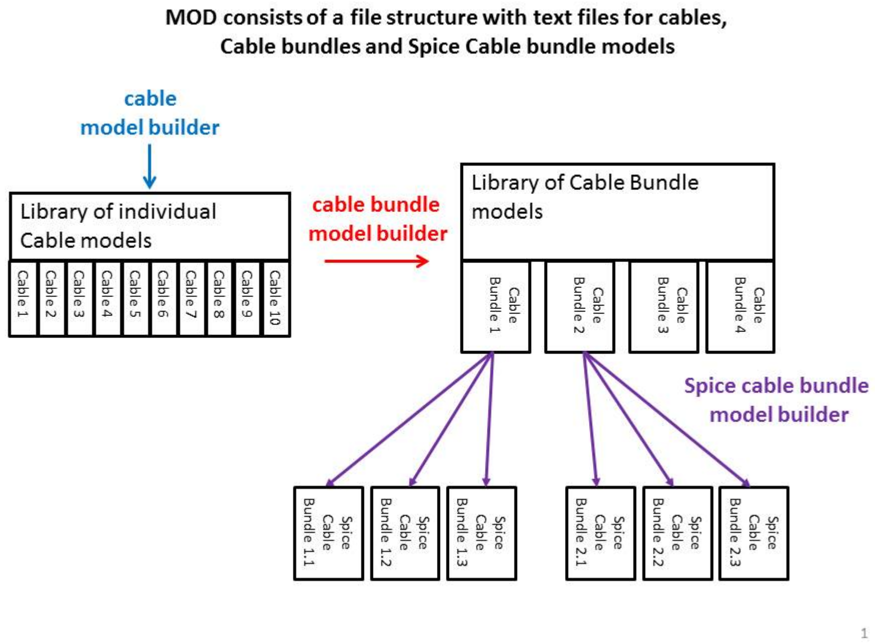


Figure 5.1 Structure of the library of cable models (MOD)

5.1 Example of a Library of cable models

An example of the library of cable models generated in the running of the test cases is shown in figure 5.1 below.

```
MOD
|-- BUNDLE_MODELS
|   |-- 2_wire
|   |   |-- 2_wire_ac.lib
|   |   `-- 2_wire_trans.lib
|   |-- 2_wire.bundle
|   |-- two_wires_over_ground.bundle
|   |-- wire_over_ground
|   |   `-- wire_over_ground_ac.lib
|   `-- wire_over_ground.bundle
`-- CABLE_MODELS
    `-- single_wire.cable
```

5.2 Using MOD with different spice versions

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 buliding 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 NAME	Plot the results of the spice validation model and analytic solution to x11 terminal
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 x11 terminal
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 file**s 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**). c. the analytic solution of the validation configuration (**analytic_solution.dat**)
5. run **ngspice** on the spice validation model to produce the validation data (**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 this chapter the inputs and outputs to these processes are outlined. The datail of the input file formats 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. 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 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

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. These flags consist of text commands. The available flags and their effect is as follows:

   'verbose' output detailed summary of the software operation and calculation results.

   'use_laplace' use the numerical Laplace solver to calculate inductance and capacitance matrices where appropriate (i.e. where an exact analytic solution is not available.) By default, approximate analytic formulae are used.

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

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

   'Laplace_surface_mesh_constant' This parameter determines the number of finite element edges on a conductor surface. The number of elements on a cylindrical conductor of radius r is $\frac{r}{Laplace_{s}urface_{m}esh_{c}onstant}$. The default value is 3.

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 us to model the external domain and 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 ***** REFERENCE REQUIRED ***** or an approximate analytic solution ***** REFERENCE REQUIRED ***** according to flags set by the user. Losses will be based on models of skin effect appropriate for the conductor geometry.

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. These flags consist of text commands. The available flags and their effect is as follows:

'verbose' output detailed summary of the software operation and calculation results.

'use_laplace' use the numerical Laplace solver to calculate inductance and capacitance matrices where appropriate (i.e. where an exact analytic solution is not available.) By default, approximate analytic formulae are used ***** REFERENCE REQUIRED *****.

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

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

   '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 ground plane point), bundle_size. The outer boundary is defined as a circle of radius $bundle_size*Laplace_boundary_constant$. The default value is 3.

   'Laplace_surface_mesh_constant' This parameter determines the number of finite element edges on a conductor surface. The number of elements on a cylindrical conductor of radius r is $\frac{r}{Laplace_surface_mesh_constant}$. The default value is 3.

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 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. We 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. These flags consist of text commands. The available flags and their effect is as follows:

   'verbose' output detailed summary of the software operation and calculation results.

   'use_xie' use Xie's model for incident field excitation of shielded cables.

   'no_s_xfer' For first order frequency dependent models (transfer impedance, propagation correction) we may use a passive circuit implementation for frequency dependent transfer functions instead of s-domain transfer functions. This may be of use in Ngspice models which fail to run.

9. Constants to be changed from their default values. 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:

   '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.

   '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 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).

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.

Chapter 8

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.

8.0.2 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**

8.0.3 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 wondows, 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.

8.0.4 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.

8.0.5 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 converegence 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 [5]

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] Christopher Smartt, David Thomas, Steve Greedy, Jaco Verpoorte, Jesper Lansink Rotgerink and Harmen Schippers *Theory Manual, Open Souce Cable Models for EMI Simulations*

[2] Christopher Smartt, David Thomas, Steve Greedy, Jaco Verpoorte, Jesper Lansink Rotgerink and Harmen Schippers *Validation Test Cases, Open Souce Cable Models for EMI Simulations*

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

[4] 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.

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